# Hasło maskowane

Projekt zaliczeniowy

## Wprowadzenie

Mechanizm hasła maskowanego pozwala uwierzytelnić użytkownika za pomocą wybranych znaków jego hasła.

Wprowadź hasło:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | | | | | | |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |

Wprowadzenie wybranych znaków ma utrudnić atakującemu przechwycenie całego hasła i zalogowanie się do systemu w imieniu użytkownika. Można by pomyśleć, że mechanizm ten zawsze podnosi poziom bezpieczeństwa, ale nic bardziej mylnego...[1,2,3]

*[...] hasła maskowane (poza tym, że są niesamowicie irytujące) mogą powodować obniżenie bezpieczeństwa (użytkownicy zapisują hasła, aby policzyć litery), a już na pewno nie są blokadą nie do przejścia przez atakujących. Podczas włamania do Plus Banku (korzystającego z haseł maskowanych), przestępcy je obeszli, prosząc użytkownika o logowanie się kilka razy, za każdym z inną maską.*[4]

## Uwierzytelnienie użytkownika

Najprostszym sposobem sprawdzenia poprawności wpisanych elementów hasła, byłoby porównanie ich ze znakami zapisanymi w jawnej postaci wewnątrz bazy danych aplikacji. Takie podejście ma jednak dość poważną wadę. W przypadku włamania atakujący mógłby poznać hasła użytkowników, a tego należy unikać.

1 https://wampir.mroczna-zaloga.org/archives/1058-hasla-maskowane-znowu.html
2 https://wampir.mroczna-zaloga.org/archives/235-o-haslach-maskowanych.html
3 http://krzysztofjelonek.net/hasla-maskowane-bledy-implementacyjne-polskiego-banku/
4 https://niebezpiecznik.pl/post/bz-wbk-nie-przestaje-zaskakiwac/

*Notatki:*

.....

.....

Lepsze byłoby użycie do tego celu geometrii matematycznej. Wykorzystując losowo wybraną prostą na płaszczyźnie, współrzędne poszczególnych punktów leżących na tej prostej, a także miejsce przecięcia z osią Y można opracować bezpieczny sposób weryfikacji hasła.[5,6]

Na początku musimy ustalić alfabet, którym będzie mógł posługiwać się użytkownik, aby utworzyć hasło. Każdemu znakowi przypisujemy pewną liczbę, przykładowo:

| | |
|---|---|
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |
| E | 5 |
| + | 6 |
| ! | 7 |

Cały proces uwierzytelniania użytkownika składa się z dwóch kroków:

1. utworzenie i zapisanie w bazie danych informacji na temat hasła,
2. weryfikacja wprowadzonego przez użytkownika hasła maskowanego.

## Utworzenie i zapisanie informacji na temat hasła

Utworzenie informacji na temat hasła przebiega w następujących krokach:

1. Wybieramy dwa losowe punkty na płaszczyźnie, które utworzą nam prostą.
2. Losujemy kolejne punkty znajdujące się na tej prostej. Łączna liczba punktów (dwa poprzednio wylosowane + liczba nowych losowych punktów) powinna odpowiadać liczbie znaków w haśle użytkownika.
3. Każdemu z wylosowanych punktów przyporządkowujemy jeden znak hasła.
4. Od współrzędnej *y* każdego z punktów odejmujemy liczbową wartość znaku odpowiadającego temu punktowi.

5 https://zaufanatrzeciastrona.pl/post/kryptografia-hasel-maskowanych-czyli-magia-matematyki/
6 https://www.youtube.com/watch?v=LP9xFk38g44

*Notatki:*

.....

.....

5. Tak przygotowane współrzędne, ich powiązanie ze znakiem hasła oraz miejsce przecięcia osi Y przez prostą zapamiętujemy.

**Przykład**:

Hasło użytkownika: `AB+CE`

1. Losujemy dwa punkty i tworzymy równanie prostej:

- (5, 6); (7, 11)
- $y=\frac{5}{2}x-\frac{13}{2}$

2. Losujemy kolejne punkty znajdujące się na prostej. Potrzebujemy pięciu, dwa już mamy.

- (3, 1)
- (9, 16)
- (13, 26)

3. Każdemu punktowi przypisujemy jeden znak hasła:

- (5, 6) → A
- (7, 11) → B
- (3, 1) → +
- (9, 16) → C
- (13, 26) → E

4. Odejmujemy wartość znaku od współrzędnej *y*.

- (5, 5)
- (7, 9)
- (3, -5)
- (9, 13)

*Notatki:*

.....

.....

- (13, 21)

5. Obliczamy miejsce przecięcia osi Y. $-\frac{13}{2}$

Zapamiętujemy przygotowane współrzędne oraz miejsce przecięcia osi Y.

## Weryfikacja wprowadzonego hasła maskowanego

Do weryfikacji wprowadzonego hasła maskowanego potrzebujemy:

- współrzędne odpowiadające każdemu znakowi hasła,
- miejsce przecięcia osi Y przez prostą.

Weryfikacja przebiega w następujących krokach:

1. Do współrzędnej Y każdego z punktów dodajemy wartość liczbową odpowiedniego znaku hasła.
2. Wybieramy dwa dowolne punkty i tworzymy na ich podstawie równanie prostej.
3. Sprawdzamy, czy pozostałe punkty należą do utworzonej prostej i czy przecina ona oś Y w zapamiętanym punkcie. Wpisane znaki są poprawne, gdy oba warunki są spełnione.

**Przykład**:

Zapamiętane dane na temat hasła:

- miejsce przecięcia osi Y: $-\frac{13}{2}$
- współrzędne: (5, 5); (7, 9); (3, -5); (9, 13); (13, 21)

Pytamy użytkownika o losowe znaki hasła:

| | B | + | | E |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

1. Do współrzędnej Y każdego z punktów zapytanego hasła dodajemy wartość liczbową odpowiedniego znaku.

*Notatki:*

.....

.....

- (7, 9) → (7, 11)
- (3, -5) → (3, 1)
- (13, 21) → (13, 26)

2. Na podstawie dowolnych dwóch punktów tworzymy równanie prostej, wybieram pierwsze dwa:

- (7, 11)
- (3, 1)

Równanie prostej: $y=\frac{5}{2}x-\frac{13}{2}$

3. Sprawdzamy, czy:

- punkt (13, 26) spełnia równanie prostej → **TAK**
- miejsce przecięcia utworzonej prostej zgadza się z zapamiętanym miejscem przecięcia → **TAK**

Oba warunki są spełnione, więc hasło jest poprawne.

## Opis projektu

Jako „Specjalista ds. bezpieczeństwa IT” pewnego dużego banku masz za zadanie napisać aplikację weryfikującą wprowadzone przez użytkownika fragmenty hasła. Twoja aplikacja będzie komunikowała się z innymi częściami systemu za pomocą standardowego wejścia i wyjścia[7], dlatego Twój program musi ściśle przestrzegać dostarczonej dokumentacji (co do znaku!).

Danymi wejściowymi są następujące zestawy danych:

- napis `ALFABET`, następnie kolejne litery alfabetu wraz z odpowiadającymi im wartościami,

7 https://pl.wikipedia.org/wiki/Standardowe_strumienie

*Notatki:*

.....

.....

- napis `SPRAWDZ`, następnie unikalny identyfikator, zapamiętane miejsce przecięcia osi Y oraz współrzędne punktów, po czym znajdują się litery wprowadzonego przez użytkownika hasła wraz z numerem znaku (znaki hasła liczymy od 1),
- napis `KONIEC` informujący program, że ma zakończyć działanie.

Każdy zestaw danych to jedna linia, dane są oddzielone pojedynczym znakiem spacji. Liczby rzeczywiste zapisane będą w formacie: XX.XXX Współrzędne punktów podane będą zapisane w formacie `(x,y)`, gdzie x, y mogą być liczbami rzeczywistymi. Każdy zestaw danych (oprócz KONIEC) może w wejściu programu wystąpić więcej niż jeden raz.

Dla każdego zestawu danych program powinien wypisać oddzielony jedną spacją unikalny identyfikator oraz napis:

- `'Ok'` – gdy hasło jest poprawne,
- `'NotOk'` – gdy hasło jest niepoprawne.

Jeżeli dane wejściowe nie są poprawne (np. współrzędne zapisane w błędnym formacie, niedozwolone znaki, napis zamiast liczby, brakujące dane), zamiast powyższych informacji na standardowe wyjście powinien zostać wypisany napis `'BLAD'`.

**Przykład 1 działania programu**:

```
ALFABET A 1 B 2 C 3 D 4 E 5 + 6 ! 7
SPRAWDZ USER_X2C1 -6.5 (5,5) (7,9) (3,-5) (9,13) (13,21) B 2 + 3 E 5
KONIEC
```

```
USER_X2C1 Ok
```

**Przykład 2 działania programu** *(brak litery E w alfabecie)*:

```
ALFABET A 1 B 2 C 3 D 4
SPRAWDZ USER_X2C1 -6.5 (5,5) (7,9) (3,-5) (9,13) (13,21) B 2 + 3 E 5
KONIEC
```

```
BLAD
```

*Notatki:*

.....

.....

**Przykład 3 działania programu**:

```
ALFABET A 1 B 2 C 3 D 4 E 5 + 6 ! 7
SPRAWDZ USER_X2C1 -6.5 (5,5) (7,9) (3,-5) (9,13) (13,21) B 2 + 3 E 5
SPRAWDZ USER_VAY1 -11 (1,2) (2,11.2) (3,0) (9,13) A 1 A 2 C 4
ALFABET A 1 C 4 D 3 D 0
SPRAWDZ USER_FF13
SPRAWDZ ADM_33112 5.1 (1,4) (13,5) (53,1)
KONIEC
```

```
USER_X2C1 Ok
USER_VAY1 NotOk
BLAD
BLAD
```

## Punktacja

Obowiązująca punktacja (w nawiasach podano możliwe do zdobycia liczby punktów):

- (0pkt albo 2pkt) - program działa poprawnie dla pierwszego podanego w treści zadania przykładu,
- (0pkt albo 4pkt) - program działa poprawnie dla drugiego podanego w treści zadania przykładu,
- (od 0pkt do 12pkt) - program działa poprawnie dla poprawnie wprowadzonych danych,
- (od 0pkt do 12pkt) - program poprawnie rozpoznaje błędnie wprowadzone dane wejściowe.

*Powodzenia!*

*Notatki:*

.....

.....