# Laboratorium z kryptografii

Zajęcia 13-14: Funkcje skrótu i podpisy cyfrowe

## 1 Funkcje skrótu

W procesie przetwarzania cyfrowych informacji bardzo łatwo spotkać się sytuacją, w której niezbędnym jest aby utworzyć stosunkowo krótki „odcisk palca“ dużo dłuższego dokumentu. Taki „odcisk palca”, czyli pewnego rodzaju unikatową informację, która pozwoli na jednoznaczne ( z bardzo dużym prawdopodobieństwem) przypisanie jej do danego dokumentu uzyskuje się przy wykorzystaniu tzw. funkcji (algorytmów) skrótu h: $\Sigma^X \to \Sigma^Y$ (ang. hash functions/algorithms ). Dobra funkcja skrótu powinna posiadać następujące własności:

1. Dla dowolnego tekstu $m_{in}$, otrzymanie $h(m_{in})$ jest zadaniem obliczeniowo łatwym.
2. Wynik $h$ jest stałej długości - funkcja skrótu zwraca ciąg znaków $m_{out} \in Y$ o stałej długości dla dowolnego wejściowego tekstu $m_{in} \in X$ o do dowolnej długości. (np. $|Y| = 256$ bitów )
3. Odporność na kolizje - znalezienie dwóch znaczących tekstów $m_1 \in \Sigma^{X_1}, m_2 \in \Sigma^{X_2}$ takich, że $h(m_1) = h(m_2)$ musi być zadaniem obliczeniowo „niewykonalnym“.
4. Jednokierunkowość - dla dowolnego skrótu $h(m_{in})$, odtworzenie dokumentu $m_{in}$ musi być zadaniem obliczeniowo „trudnym”.

   Przykładowe funkcje skrótu:

| Nazwa | Długość skrótu | Względna szybkość |
|---|---|---|
| MD4 | 128 | 1.00 |
| MD5 | 128 | 0.68 |
| RIPEMD-128 | 128 | 0.39 |
| SHA-1 | 160 | 0.28 |
| RIPEMD-160 | 160 | 0.24 |

Tabela 1: Przykładowe funkcje skrótu, długości zwracanych przez nie skrótów oraz względna szybkość.

### 1.1 Funkcja skrótu JHA

Funkcja skrótu $JHA$ jest algorytmem utworzonym na potrzeby dydaktyczne przez prof. J. Holdena [1]. Dla zadanego tekstu $m$ niech

- $n_1(m)$ liczba samogłosek j. angielskiego ({a,e,i,u,o,A,E,I,U,O}) występujących w tekście $m$
- $n_2(m)$ liczba spółgłosek j. angielskiego ({b,c,d,...,x,y,z,B,C,D,...,X,Y,Z}) występujących w tekście $m$
- $SP(m)$ Liczba spacji występujących w teście $m$

wtedy $JHA(m, p, q)$ zdefiniujemy poprzez

$$JHA(m, p, q) = q^{7*n_1(m)-3*n_2(m)+(SP(m))^2} \mod p \tag{1}$$

Uwaga! Wyrażenie postaci $(a^{-k}) \mod p$ jest liczbą naturalną i oznacza $k$-tą potęgę modulo $p$ elementu odwrotnego do $a$ w ciele $\mathbb{Z}_p$, tzn:

$$a^{-k} \mod p = b^k \mod p \tag{2}$$

$$a \cdot b = 1( \mod p) \tag{3}$$

[1] https://www.rose-hulman.edu/~holden/Preprints/jha-paper.pdf

| lp. | Tekst $m$ | $n_1(m)$ | $n_2(m)$ | $SP(m)$ | p | q | $JHA(m,p,q)$ |
|---|---|---|---|---|---|---|---|
| $m_1$ | Aaaa aAAA | 8 | 0 | 1 | 541 | 5 | 368 |
| $m_2$ | aEEE AEIU O | 9 | 0 | 2 | 101 | 5 | 54 |
| $m_3$ | Aa A ee cCc | 5 | 3 | 3 | 1 223 | 47 | 70 |
| $m_4$ | Aa ABb cde | 4 | 4 | 2 | 1 583 | 113 | 256 |
| $m_5$ | A B C d e f gh | 2 | 6 | 6 | 1 987 | 331 | 1 674 |
| $m_6$ | AaBBB BBB cdcd | 2 | 10 | 2 | 2 741 | 137 | 2 126 |
| $m_7$ | A bcdf GH IJK | 2 | 8 | 3 | 3 571 | 17 | 3 361 |
| $m_8$ | aA BBBa CDEE eee | 8 | 5 | 3 | 5 279 | 29 | 3 249 |
| $m_9$ | BbBbBb CcCcCcCc DdDdDdD | 0 | 21 | 2 | 6 997 | 53 | 4 070 |
| $m_{10}$ | ABCD abcd ABCD BBBB | 3 | 13 | 3 | 8 831 | 883 | 4 778 |
| $m_{11}$ | AAaaAA bB iuo | 9 | 2 | 2 | 12 553 | 523 | 3 194 |
| $m_{12}$ | A,B, , C?!eee iuo!DA | 8 | 3 | 3 | 27 449 | 73 | 19 032 |
| $m_{13}$ | ?? AaaA,.cdef BBBB?#@ | 5 | 7 | 2 | 127 | 7 | 64 |
| $m_{14}$ | 2Mama2 i ?Tata? :D | 5 | 5 | 3 | 29 | 7 | 7 |

Tabela 2: Przykładowe teksty oraz wartości ich skrótów.

# 2 Podpisy cyfrowe - podstawy

Podpis cyfrowy dokumentu powinien spełniać analogiczną funkcję jak podpis ręczny dokumentu papierowego, tzn. stanowić publicznie uznane oraz prawnie wiążące narzędzie pozwalające na utworzenie unikatowej informacji bezpośrednio związanej z podpisywanym dokumentem oraz podmiotem podpisującym. Każda taka unikatowa informacja powinna przede wszystkim zapewniać bezpieczeństwo ( być obliczeniowo niemożliwa do podrobienia ) oraz łatwa do sprawdzenia ( każda osoba otrzymująca taki podpis powinna w prosty sposób sprawdzić jego autentyczność ), podpis cyfrowy najczęściej powiązany jest ze zbiorem dwóch protokołów:

- Właściwego protokołu podpisu cyfrowego $SG_{private}(m)$ wiadomości $m$ przy wykorzystaniu tajnej informacji *private*.
- Protokołu weryfikacji $VER_{public}(\tilde{m})$ podpisu $\tilde{m}$ przy wykorzystaniu publicznej informacji *public* (np. składowanej w tzw. publicznym rejestrze kluczy (PRK) stanowiącym źródło informacji powszechnie wiarygodnych i dostępnych).

Ponieważ podpis cyfrowy, jak każda informacja cyfrowa, może być z łatwością powielany, przesyłany i przetwarzany po jego wykonaniu ( często bez wiedzy osoby podpisującej ), protokół podpisu $SG_{private}$ oraz weryfikacji $VER_{public}$ powinny m.in. spełniać poniższe kryteria bezpieczeństwa

1. Łatwość wykonania podpisu - przy znajomości tajnej informacji *private* podpisanie $SG_{private}(m)$ dokumentu $m$ jest zadaniem obliczeniowo łatwym
2. Łatwość weryfikacji podpisu - przy znajomości publicznej informacji *public* wykonanie protokołu sprawdzenia autentyczności $VER_{public}(\tilde{m})$ podpisu $\tilde{m}$ jest zadaniem obliczeniowo łatwym
3. Odporność na podstawienia - znalezienie dwóch par $\{(private_1, m_1), (private_2, m_2)\}$ takich, że $SG_{private_1}(m_1) = SG_{private_2}(m_2)$ jest zadaniem obliczeniowo niewykonalnym.
4. Odporność na podrobienia - utworzenie podrobionego podpisu $\tilde{m}$ takiego, że protokół weryfikacji $VER_{public}(\tilde{m}$ potwierdza autentyczność wiadomości jest zadaniem obliczeniowo niewykonalnym.

# 3 Digital Signature Standard (DSS)

DSA oparty jest na zmodyfikowanej wersji podpisu cyfrowego opartego o algorytm ElGamal. W swojej pełnej wersji algorytm DSS można podzielić na trzy etapy.

**Etap I** Generacja publicznie dostępnej informacji ( w PRK ) pozwalającej dowolnej osobie na sprawdzenie wiarygodności podpisu.

1. wybór dużej liczby pierwszej p ( sugerowana liczba pierwsza powinna być długości co najmniej 512 bitów! )
2. wybór 160 bitowej liczby pierwszej q dzielącej liczbę p-1

3. wybór liczby naturalnej g, będącej q-tym pierwiastkiem modulo p, tzn. spełniającej:

$$g^q = 1 \mod p \tag{4}$$
$$\forall_{\alpha<q} \quad g^\alpha \neq 1 \mod p \tag{5}$$

4. Wybór tajnej liczby naturalnej $k < q$
5. Obliczenie klucza publicznego $g^k \mod p$ i zdeponowanie czwórki $(g^k, g, p, q)$ w publicznym rejestrze (PRK)

**Etap II** Podpis dokumentu.

1. wybór ( jednorazowej ) liczby naturalnej $r < q$ oraz obliczenie liczby $x$ zadanej przez:

$$x = \left(g^k \mod p\right) \mod q \tag{6}$$

2. dla wiadomości $m \in \mathbb{Z}_q^*$ obliczenie liczby $y$ zadanej przez:

$$y = r^{-1}\left(m + k \cdot x\right)( \mod q) \tag{7}$$

3. Podpisem wiadomości $m$ jest para $(x, y)$ tzn:

$$s = SG_k(m) = (x, y) \tag{8}$$

**Etap II** Weryfikacja podpisu

1. Dla otrzymanego podpisu $\tilde{s} = (\tilde{x}, \tilde{y})$ oraz wiadomości $\tilde{m}$, obliczenie

$$\alpha = \tilde{m}\tilde{y}^{-1}( \mod q) \tag{9}$$
$$\beta = \tilde{x}\tilde{y}^{-1}( \mod q), \tag{10}$$

gdzie $q$ pochodzi z publicznego rejestru podpisów.

2. Autentyczność podpisu sprawdzana jest poprzez warunek (???). Jeżeli (???) jest prawdą to para $(\tilde{x}, \tilde{y}) = (x, y)$ jest autentycznym podpisem wiadomości $\tilde{m}$.

$$VER(\tilde{m}, \tilde{s}) = \left(\tilde{x} \stackrel{?}{=} \left(g^\alpha \cdot (g^k)^\beta \mod p\right) \mod q\right) \tag{11}$$

Wartości $(g^k, g, p, q)$ pochodzą z publicznego rejestru kluczy.

| lp. | Tekst $m$ | p | q | $JHA$ | g | k | r | PRK | Podpis |
|---|---|---|---|---|---|---|---|---|---|
| $m_1$ | Aaaa aAAA | 541 | 5 | 368 | 140 | 3 | 4 | (48, 140, 541, 5) | (3, 3) |
| $m_2$ | aEEE AEIU O | 101 | 5 | 54 | 84 | 2 | 3 | (87, 84, 101, 5) | (2, 1) |
| $m_8$ | aA BBBa CDEE eee | 5 279 | 29 | 3 249 | 2 160 | 23 | 17 | (1 186, 2 160, 5 279, 29) | (26, 25) |
| $m_9$ | BbBbBb CcCcCcCc DdDdDdD | 6 997 | 53 | 4 070 | 1 001 | 48 | 33 | (2 328, 1 001, 6 997, 53) | (49, 34) |
| $m_{10}$ | ABCD abcd ABCD BBBB | 8 831 | 883 | 4 778 | 6 275 | 700 | 500 | (5 343, 6 275, 8 831, 883) | (45, 839) |

Tabela 3: Przykładowe wyniki DSS.

# 4 Zadania

1. Dla zadanych w konsoli liczb $(p, g, k, r)$ (liczba pierwsza $p$ <32 000) zaimplementować program dokonujący skrótu wiadomości $h = JHA(m, p, q)$ oraz podpisujący otrzymany skrót algorytmem DSS.
   **Wytyczne implementacji:**
   - Wszystkie zmienne wejściowe (tzn. $p, g, k$ oraz $r$) powinny być podawane w konsoli!
   - Program powinien automatycznie obliczać $q$ (dla skrótu oraz podpisu!) jako największą liczbę pierwszą dzielącą $p - 1$ (np. wykorzystując algorytm fermata).
   - Program powinien automatycznie dokonywać kontroli błędów dla liczb wejściowych $(p, g, k, r)$, w razie wykrycia błędu wyświetlać odpowiedni komunikat i prosić o nową wartość dla zmiennej.
   - Program powinien pobierać tekst $m$ z pliku ( oceniany plik: tekst_dlugi.txt oraz pomocnicze pliki testowe tekst_krotkix.txt dostępne są na e-nauczaniu ).
   - Program ma wyświetlać: skrót $h$ wiadomości $m$, podpis $s$ wiadomości $h$ oraz utworzony publiczny rejestr klucza $(g^k, g, p, q)$!

# 5 Punktacja

- 1 punkt - wczytywanie $(p, g, k, r)$ w konsoli i tekstu $m$ z pliku
- 3 punkty - poprawnie zaimplementowana kontrola błędów
- 3 punkty - poprawnie otrzymany skrót wiadomości
- 3 punkty - poprawnie otrzymany rejestr klucza publicznego oraz podpis skrótu wiadomości