Digital Signature Algorithm

Filip Cebula, 151410

10 stycznia 2025

1 Wstęp

DSA to algorytm kryptograficzny, który służy do generowania i weryfikacji cyfrowych podpisów. Algorytm ten opiera się na problemie logarytmu dyskretnego, co sprawia, że jest trudny do złamania. DSA jest wykorzystywany, do potwierdzania autentyczności i integralności przesyłanych informacji. Algorytm składa się z trzech części: generacji kluczy, podpisywania i weryfikacji podpisu.

2 Generacja kluczy

Generacje kluczy możemy podzielić na dwie fazy. Pierwszą jest dobór odpowiednich parametrów algorytmu, a drugą jest generacja klucza prywatnego i publicznego dla użytkownika. Klucz publiczny powinien być przekazany do osób, które będą odbierać nasze wiadomości, natomiast klucz prywatny, jak nazwa wskazuje, nie powinien być udostępniany nikomu.

2.1 Pseudokod

Algorithm 1 Algorytm generacji kluczy

Require: **L**, **N** ▷ Rekomendowana wartość L to 2048 lub 3072
Ensure: **N** jest równe długości wyjścia funkcji haszującej (np. SHA256)

Wybieramy liczbę pierwszą **q** o rozmiarze **N** bitów.
Wybieramy liczbę pierwszą **p** o rozmiarze **L** bitów taką, że $q|p-1$.
repeat
    Wybieramy losową liczbę całkowitą **h**, taką że $h \in [2; p-2]$
    $g \leftarrow h^{(p-1)/q} \pmod p$
until $g \neq 1$
Wybieramy losową liczbę całkowitą **x**, taką że $x \in [1; q-1]$
$y \leftarrow g^x \pmod p$
return (p,q,g) ▷ Parametry algorytmu DSA
return x ▷ Klucz prywatny
return y ▷ Klucz publiczny

2.2 Przykład

1. Na potrzeby przykładu użyjemy mniejszych rozmiarów liczb, tj. **L=16** i **N=4**.
2. Wybieramy liczbę **q=13**.
3. Wybieramy liczbę **p=57773**.
4. Wybieramy losową liczbę całkowitą **h**, z przedziału $[2; 57771]$. W naszym przykładzie weźmy liczbę **h=37154**.
5. Obliczamy liczbę **g**, kożystając z szybkiego potęgowania modulo. W naszym przypadku **g=45887**.
6. Wybieramy losową liczbę całkowitą **x** z przedziału $[1; 12]$. W naszym przypadku **x=4**. Liczba **x** to klucz prywatny.
7. Obliczamy liczbę **y**. W naszym przypadku **y=57516**. Liczba **y** to nasz klucz publiczny.

3 Generacja podpisu

Algorytm DSA pozwala nam na stworzenie podpisu do danej informacji, dzięki czemu odbiorca informacji może zweryfikować źródło, z którego pochodzi informacja, oraz czy informacja nie została w żaden sposób zmodyfikowana, zanim została odebrana.

Podpisywanie wiadomości jest najbardziej czasochłonną operacją w algorytmie DSA, jest to głównie spowodowane obliczeniem wartości **r** i k^{-1}. Jednak wartość **r** nie jest zależna od wiadomości, a jedynie od parametrów algorytmu, może być więc obliczona przed procesem podpisywania.

Należy także pamiętać, że wybrana przez nas liczba losowa **k**, powinna być unikalna dla każdej wiadomości podpisanej danym kluczem prywatnym. W wypadku, gdy liczba **k** powtórzy się dla dwóch różnych wiadomości, możliwe będzie obliczenie klucza prywatnego **x**, użytego do podpisania informacji. Aby tego uniknąć zaleca się generowanie **k** w sposób deterministyczny na podstawie klucza prywatnego **x** i haszu podpisywanej informacji **H(M)**.

3.1 Pseudokod

Algorithm 2 Algorytm generacji podpisu

Require: M ▷ Informacja, którą chcemy podpisać
Require: p,**q**,**g** ▷ Parametry algorytmu DSA
Require: x ▷ Klucz prywatny
Niech **H(M)** oznacza wynik funkcji haszującej **H** na informacji **M**
repeat
    Wybieramy losową liczbę całkowitą **k**, taką że $k \in [1; q-1]$
    $r \leftarrow (g^k \pmod p) \pmod q$
    $s \leftarrow (k^{-1}(H(M) + xr)) \pmod q$
until $s \neq 0 \wedge r \neq 0$
return (r,s) ▷ Wygenerowany podpis

3.2 Przykład

Do pokazania procesu podpisywania informacji, użyjemy obliczonych wcześniej przez nas parametrów **(p,q,g)**, oraz kluczy **x** i **y**. Przyjmujemy, że chcemy podpisać informacje **M**, oraz że wartość **H(M)=17**.

1. Wybieramy losową liczbę **k** z przedziału [1;12], np. **k=4**.
2. Obliczamy **r**. $r = (45887^4 \pmod 5 7773) \pmod 1 3 = 4$.
3. Obliczamy $4^{-1} \pmod 1 3 = 10$
4. Obliczamy **s**. $s = (10 * (17 + 4 * 4)) \pmod 1 3 = 330 \pmod 1 3 = 5$.
5. Otrzymana para liczb **(r,s)** to nasz podpis.

4 Weryfikacja podpisu

4.1 Pseudokod

Algorithm 3 Algorytm weryfikacji podpisu

Require: H(M) ▷ Hasz odebranej informacji
Require: r, **s** ▷ Para liczb tworzących podpis
Require: p,**q**,**g** ▷ Parametry algorytmu DSA
Require: y ▷ Klucz publiczny
if $0 < r < q \wedge 0 < s < q$ **then**
  $w \leftarrow s^{-1} \pmod q$
  $v_1 \leftarrow H(M) * w \pmod q$
  $v_2 \leftarrow r * w \pmod q$
  $v \leftarrow (g^{v_1} y^{v_2} \pmod p) \pmod q$
  return $v = r$ ▷ Podpis jest poprawny, jeżeli wartość tego wyrażenia jest prawdziwa.
else
  return Niepoprawny format podpisu
end if

4.2 Przykład

Jak w poprzednim punkcie, użyjemy wcześniej wyliczonych przez nas wartości, aby zweryfikować poprawnosc podpisu. Wiemy, że warunek $0 < r < q0 < s < q$ jest spełniony więc ten krok możemy pominąć.

1. Obliczamy wartość **w**. $w = 5^{-1} \pmod 1)3 = 8$
2. Obliczamy wartość v_1. $v_1 = 17 * 8 \pmod 1)3 = 6$
3. Obliczamy wartość v_2. $v_2 = 4 * 8 \pmod 1)3 = 6$
4. Obliczamy wartość **v**. $v = (45887^6 * 57516^6 \pmod 5)7773) \pmod 1)3 = 57516 \pmod 1)3 = 4$
5. Otrzymaliśmy wartość $v = 4$. $v = r$ z czego wynika, że podpis jest poprawny.

4.3 Poprawność weryfikacji

Aby weryfikacja była poprawna musimy udowodnić że dla każdego, poprawnie wygenerowanego, **r** i **v**, odpowiadającemu temu samemu podpisowi $r = v$.

$$r = v$$
$$(g^k \pmod p) \pmod q = (g^{v_1} y^{v_2} \pmod p) \pmod q$$

Wiemy także, że:

$$w = s^{-1} \pmod q$$
$$s \equiv k^{-1}(H(M) + xr) \pmod q$$
$$y \equiv g^x \pmod p$$

Po przekształceniu:

$$k \equiv s^{-1}(H(M) + xr) \pmod q$$
$$k \equiv H(M)w + xrw \pmod q$$

Wiemy także, że **g** jest pożądku **q**, ponieważ

$$g = h^{(p-1)/q} \pmod p$$
$$g^q = h^{(p-1)} \pmod p$$

Co na podstawie małego twierdzenia Fermata oznacza

$$g^q \equiv h^{(p-1)} \pmod p \equiv 1$$

Więc

$$\begin{aligned} g^k &\equiv g^{H(M)w+xrw} \\ &\equiv g^{H(M)w} g^{xrw} \\ &\equiv g^{H(M)w} y^{rw} \pmod p \end{aligned}$$

Pamiętając, że

$$v_1 \equiv H(M) * w \pmod q$$
$$v_2 \equiv r * w \pmod q$$

Po podstawieniu do wcześniejszego wzoru:

$$g^k \equiv g^{H(M)w} y^{rw} \pmod p \equiv g^{v_1} y^{v_2} \pmod p$$
$$r = (g^k \pmod p) \pmod q = (g^{v_1} y^{v_2} \pmod p) \pmod q = v$$

Co kończy dowód.

5 Przykładowy program