

Projekt aplikacji podpisu cyfrowego.

Aleksandra Nowak, Tymon Świtalski, Piotr Sobczyk, Jakub Lis

20 czerwca 2023

1 Generowanie kluczy

Generując parę kluczy — publiczny i prywatny, należy wybrać dwie duże liczby pierwsze, p oraz q . Wybieramy też liczbę naturalną $e > 1$ spełniającą równanie:

$$NWD(e, (p-1)(q-1)) = 1 \quad (1)$$

Częste wybory to 3, 17 lub 65537. Im większa wartość e , tym trudniej złamać zaszyfrowane wiadomości, ale jednocześnie tym dłużej czasu zajmować będzie szyfrowanie i odszyfrowywanie. Wielkość liczb p oraz q również wpływa na siłę algorytmu. Podsumowując — im większe liczby, tym lepsze szyfrowanie. Na podstawie wybranych liczb, wyliczamy: $N = pq$ oraz publikujemy parę (N, e) jako nasz klucz publiczny. Aby wyliczyć klucz prywatny, należy zastosować rozszerzony algorytm Euklidesa do liczb e oraz $(p-1)(q-1)$, wyznaczając stałą d spełniającą równanie:

$$e \cdot d = 1 \bmod (p-1)(q-1) \quad (2)$$

Ponieważ e oraz $(p-1)(q-1)$ są względnie pierwsze (tak zostało wybrane e), rozwiązanie opiera się na tożsamości Bézout’a. W szczególności należy rozwiązać równanie diofantyczne liniowe:

$$d \cdot e + c \cdot (p-1)(q-1) = 1 \quad (3)$$

ze względu na (d, c) . Rozwiązanie to istnieje i znacząca jest jedynie jedna ze współrzędnych rozwiązania, ponieważ druga zniknie, gdy stronami na powyższe równanie zastosuje się operację *modulo* $(p-1)(q-1)$. Trójka liczb (d, p, q) stanowi klucz prywatny.

2 Podpis cyfrowy-czym jest i jak go skonstruować?

To matematyczny sposób sprawdzenia autentyczności dokumentów i wiadomości elektronicznych. Poprawny podpis oznacza, że wiadomość pochodzi od właściwego nadawcy, który nie może zaprzeczyć faktowi jej nadania, oraz że wiadomość

nie została zmieniona podczas transmisji. Dyspozycja dokumentem D oraz kluczem prywatnym (d, p, q) pozwala za pomocą wybranej kryptograficznej funkcji skrótu H (w tym przypadku MD_4) wyliczyć sumę kontrolną dokumentu:

$$m = H(D) \quad (4)$$

Następnie wyliczając podpis s i stosując:

$$s = m^d \bmod p \cdot q = D_{(d,p,q)}(m) \quad (5)$$

Do odbiorcy zostaje przesłany dokument D oraz podpis s . Chcąc sprawdzić, czy dokument D z podpisem s jest rzeczywiście napisany przez osobę o kluczu publicznym (N, e) , należy sprawdzić, czy suma kontrolna dokumentu jest równa odkodowanej sumie kontrolnej:

$$H(D) = s^e \bmod N \quad (6)$$

inaczej:

$$H(D) = E_{(N,e)}(s) \quad (7)$$

Jeśli równanie to jest prawdziwe, tzn. że dokument D został wysłany przez osobę posługującą się kluczem publicznym (N, e) . Pamiętając, że operacje szyfrowania i odszyfrowania spełniają równanie:

$$m = D_{(d,p,q)}(E_{(N,e)}(m)) \quad (8)$$

upraszczając:

$$m = \text{odszyfruj}(\text{szyfruj}(m)) \quad (9)$$

Co działa w obie strony:

$$m = E_{(N,e)}(D_{(d,p,q)}(E_{(N,e)}(m))) \quad (10)$$

upraszczając:

$$m = \text{szyfruj}(\text{odszyfruj}(m)) \quad (11)$$

Jest to konieczne, aby można było w zaproponowany powyżej sposób przeprowadzić weryfikację podpisu. W pierwszym kroku autor wiadomości generuje podpis, stosując swój klucz prywatny za pomocą algorytmu odszyfrowywania, a osoba weryfikująca stosuje klucz publiczny autora i algorytm szyfrowania. Zakładając, że klucze są poprawne, czyli zachodzi:

$$m = D_{(d,p,q)}(E_{(N,e)}(m)) \quad (12)$$

Oznacza to, że:

$$m = (m^e \bmod N)^d \bmod p \cdot q \quad (13)$$

przy czym, ponieważ: $N = p \cdot q$:

$$m = (m^e \bmod N)^d \bmod N \quad (14)$$

Korzystając z własności operacji *modulo* mówiącej, że:

$$A^B \bmod C = (A \bmod C)^B \bmod C \quad (15)$$

mamy:

$$m = (m^e)^d \bmod N \quad (16)$$

Z własności potęgi otrzymujemy:

$$m = (m^d)^e \bmod N \quad (17)$$

i ponownie stosując własność potęgowania modulo:

$$m = (m^d \bmod N)^e \bmod N \quad (18)$$

Ponieważ $N = pq$, ostatecznie otrzymujemy:

$$m = (m^d \bmod p \cdot q)^e \bmod N \quad (19)$$

czyli:

$$m = E_{(N,e)}(D_{(d,p,q)}(m)) \quad (20)$$

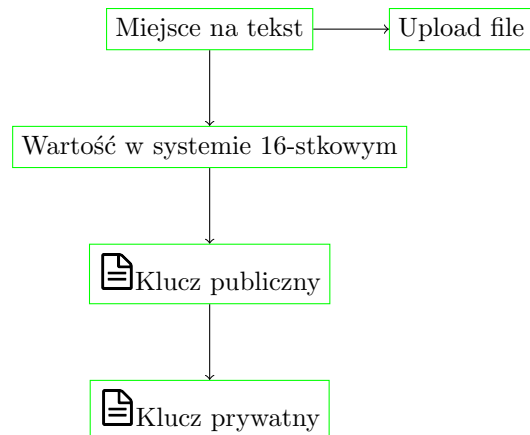
Więcej informacji można znaleźć [dAG].

3 Jak powinna wyglądać aplikacja pozwalająca tworzyć pary kluczy oraz podpisywać i sprawdzać podpisy?

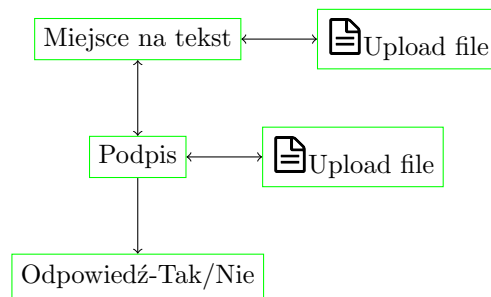
Aplikacja pozwalająca tworzyć parę kluczy i zajmująca się podpisem cyfrowym, powinna sprawdzać z każdym podpisem certyfikat i klucz publiczny osoby podpisującej, który potwierdza tożsamość. Certyfikaty są wystawiane przez urząd certyfikacji. Na ogół certyfikat jest ważny przez rok, po upływie którego osoba podpisująca musi odnowić certyfikat podpisywania lub uzyskać nowy, aby ustanowić tożsamość-ważne, aby aplikacja pilnowała tych terminów. Przestrzegając tych zasad taki program zapewnia bezpieczeństwo (na tyle, ile to możliwe). Aplikacja powinna gwarantować, iż przesłany dokument nie został zmieniony, a osoba, która go przesłała nie może się wyprzec podpisanej zawartości. Program powinien mieć możliwość poświadczenia notarialnego, tzn. podpisy w plikach programów Word, Excel lub PowerPoint, które są oznaczone sygnaturą czasową przez bezpieczny serwer sygnatur czasowych, w niektórych okolicznościach powinny mieć ważność poświadczenia notarialnego.

4 Interfejs (szkic) naszej aplikacji

4.1 Stwórz parę kluczy



4.2 Sprawdzanie klucza



5 Diagramy klas aplikacji

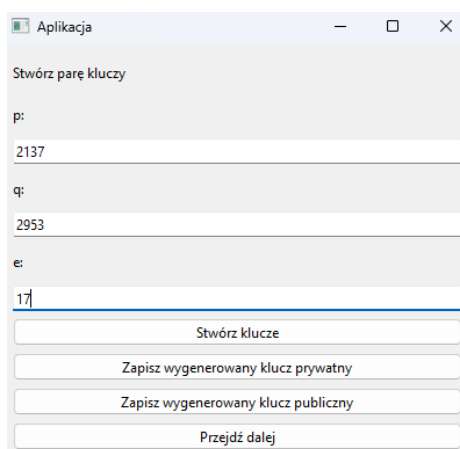
Diagramy klas powstają na podstawie dziedziczenia. Naszym "rodzicem" jest aplikacja od której dziedziczą "dzieci": dane, keys, public key.



Rysunek 1: Diagramy klas aplikacji

6 Instrukcja obsługi aplikacji

1. Krok 1: Tworzymy klucz publiczny i prywatny przy wprowadzeniu parametrów p, q i e (opisane powyżej) i klikamy "Przejdź dalej".
2. Krok 2: Wprowadzamy zaszyfrowaną wiadomość bądź plik, następnie przechodzimy dalej.
3. Krok 3: Podpisujemy zaszyfrowaną wiadomość, klikamy "Sprawdź podpis", po czym dostajemy odpowiedź od aplikacji, czy dany podpis się zgadza.



Aplikacja

Stwórz parę kluczy

p:
2137

q:
2953

e:
17

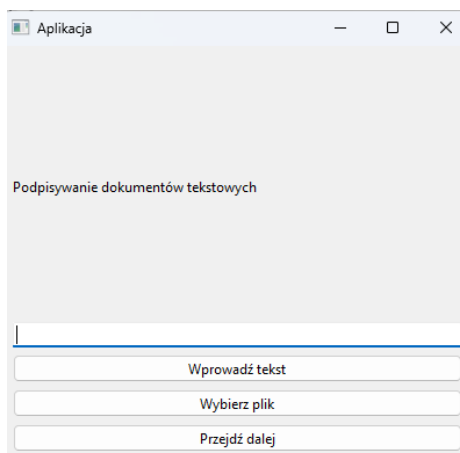
Stwórz klucze

Zapisz wygenerowany klucz prywatny

Zapisz wygenerowany klucz publiczny

Przejdź dalej

Rysunek 2: Krok 1



Aplikacja

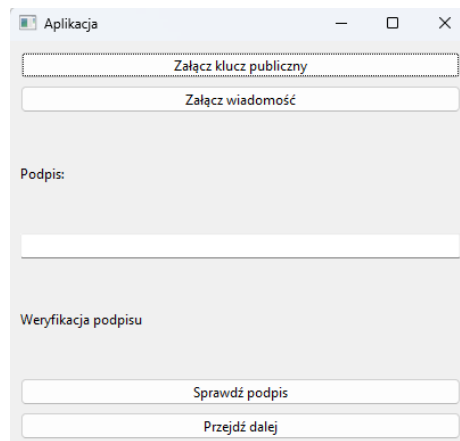
Podpisywanie dokumentów tekstowych

Wprowadź tekst

Wybierz plik

Przejdź dalej

Rysunek 3: Krok 2



Rysunek 4: Krok 3

7 Źródło

Literatura

[dAG] dr Andrzej Giniewicz. Wprowadzenie do algorytmu Rsa, szyfrowanie i podpis cyfrowy.