Laboratorium nr 3-4

# Badanie RSA

Termin wykonania: **do terminu laboratorium nr 5**
Liczba punktów: **6 + 4 + 2 dodatkowe**
PRK: T-L-3

## 1. Opis laboratorium

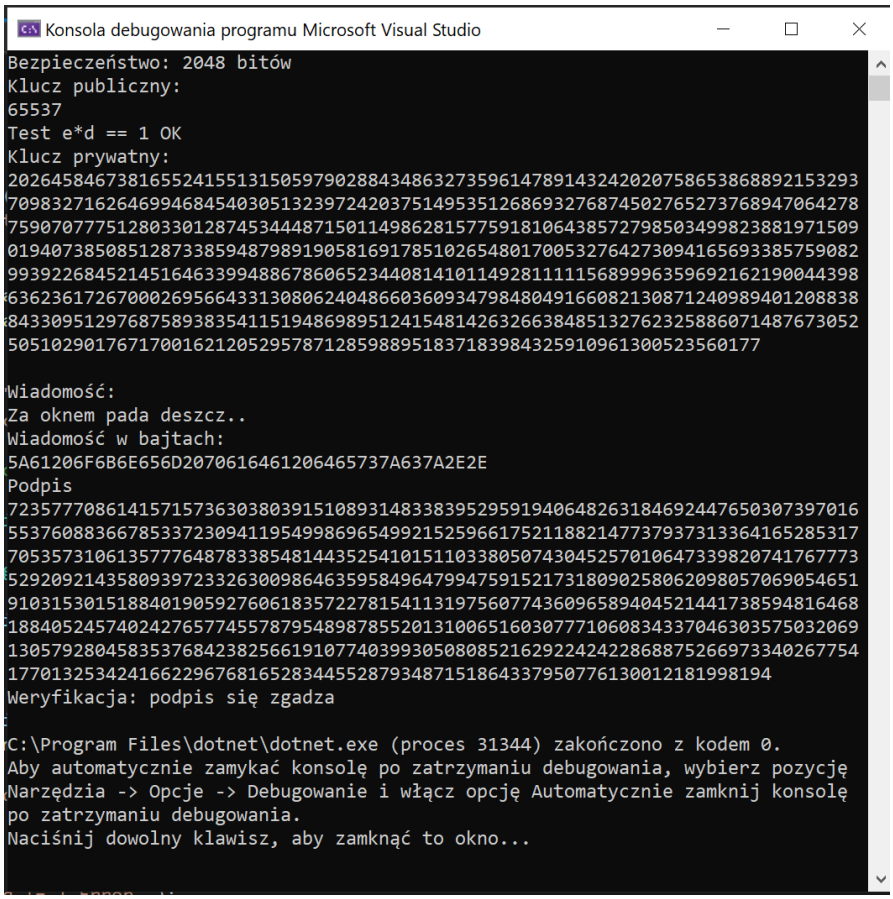
Celem laboratorium jest zbadanie kwestii zawiązanych z bezpieczną implementacją algorytmu podpisu RSA oraz zapoznanie się z modelami bezpieczeństwa schematów podpisów cyfrowych. Zadanie obejmuje implementacje, przeprowadzanie eksperymentów związanych z najbardziej znanymi atakami oraz przygotowanie analizy dotyczącej bezpiecznej implementacji RSA.

## 2. Materiały

- J. Katz, Y. Lindell, Introduction to Modern Cryptography, Second Edition, CRC Press 2015
- A. Menezes, P. van Oorschot, S, Vanstone, Kryptografia Stosowana, WNT 2005
- B. Schneier, Applied Cryptography, Second Edition, Wiley & Sons 2015
- https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf
- https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf
- https://tools.ietf.org/html/rfc3447
- https://www.cryptosys.net/pki/manpki/pki_rsaschemes.html

## 3. Zadania do wykonania

1) **Zadanie 3.1** – (1 pkt + 2 pkt do lab3) Zaimplementuj podstawowy algorytm podpisu RSA (**Plain RSA**), tj. bez użycia funkcji skrótu.
    a. *Wskazówka:* w BouncyCastle należy używać metody *compareTo* do porównywania wartości *BigInt*.
    b. *Wskazówka:* Potęgowanie dużych liczb w ciele skończonym wymaga algorytmu potęgowania modularnego (*pow mod*).
    c. *Wskazówka:* Dzielenie liczb w ciele skończonym to obliczanie ich odwrotności. Służy do tego algorytm *mod inverse*. Algorytm ten przyjmuje jako wejście liczbę dodatnią. W przypadku, gdy mamy liczbę ujemną należy dodać do niej moduł.




2) **Zadanie 3.2** – (1 pkt) Odpowiedz na następujące pytania:
   a. Wiadomość o jakiej maksymalnie długości można podpisać za pomocą tego algorytmu?
   b. Sprawdź czas wykonania operacji podpisywania i weryfikacji podpisu dla różnych wartości modułu, tj. dla 2048, 3072, 4096, 7680 bitów.
   c. Umieść w pliku z odpowiedziami przykładową liczbę składającą się z 4096 bitów, jak duża jest to liczba?
   d. Jak ustala się wartość klucza publicznego?

3) **Zadanie 4.1** – Przeprowadź następujące eksperymenty (2 + 2 pkt do lab4)
   a. Eksperyment 1 – *no-message attack* – wygeneruj losowy podpis $s$ i oblicz $m = s^e \bmod N$. Na wyjściu otrzymasz parę podpis, wiadomość $(s, m)$ wygenerowaną pomimo braku użycia klucza prywatnego.
   b. Eksperyment 2 - Adwersarz wybiera dwie wiadomość $m_1$, $m_2$ takie, że $m = m_1 m_2 \bmod N$. Następnie uzyskuje podpisy dla wiadomości $m_1$, $m_2$ odpowiednio $s_1$ i $s_2$. Adwersarz oblicza podpis dla $m$ jako $s = s_1 s_2 \bmod N$.
   c. Eksperyment 3 – *Atak na szyfrowanie z wykorzystaniem RSA.* Dany jest szyfrogram **28292467596674309017799738 75** (zapis dziesiętny). Został on zaszyfrowany algorytmem *PlainRSA* z kluczem publicznym $e=3$ oraz $N =$ **74863748466636279180898113945573168800167314349007339734664557033677222985045895878321130196223760783214379338040678233908010747773264003237620590141174028330154012139597068236121542945442426074367017838349905866915120469978361986 00**

**22403622823921817262650233787962846006970136350031500200127636665368297013349**

Odszyfruj wiadomość (uzyskane odszyfrowane liczby dziesiętne zamień na zapis szesnastkowy i z tablicy ASCII odczytaj litery).

d. (opcjonalny, **Bonus +2pkt!**) Eksperyment 4 – zademonstruj działanie ataku o nazwie *Håstad's broadcast attack* na schemat *PlainRSA* w którym ta sama wiadomość jest szyfrowana różnymi kluczami publicznymi trzykrotnie, przy czym każdy z tych kluczy jest taki sam i wynosi $e = 3$, a moduły $N_1$, $N_2$, $N_3$ są różne. Dokładny opis ataku znajdziesz w materiałach poniżej:
   i. https://cims.nyu.edu/~regev/teaching/lattices_fall_2004/ln/rsa.pdf
   ii. https://crypto.stackexchange.com/questions/6713/low-public-exponent-attack-for-rsa
   iii. https://en.wikipedia.org/wiki/Coppersmith%27s_attack

4) **Zadanie 4.2** - (1 pkt) Ulepsz schemat **PlainRSA** do wersji z załącznikiem (**RSA-FDH**), tj. algorytm ten używa funkcji skrótu (podpisywany (szyfrowany) kluczem prywatnym jest skrót z wiadomości a nie sama wiadomość). Odpowiedz dodatkowo na pytania:
   a. Jak dodanie funkcji skrótu wpływa na czas wykonania podpisywania i weryfikacji podpisu?
   b. Jak dodanie funkcji wpływa na bezpieczeństwo? Czy któryś z powyższych ataków może się nadal powieść?

5) **Zadanie 4.3** - (1 pkt) Co należy zrobić, aby implementacja RSA była bezpieczna? Odpowiedz na poniższe pytania:
   a. Co to znaczy, że schemat podpisu jest bezpieczny? Jaka jest przyjęta definicja?
   b. Wyjaśnij dlaczego ataki z pkt. 2 są możliwie do przeprowadzenia.
   c. Jaka wartość klucza publicznego należy wybrać, czy $e$ może być stałe?
   d. Jaka musi być wielkość modułu, aby uzyskać bezpieczeństwo na poziomie 256 bitów (*256 bit security*) i co znaczy tak określony poziom bezpieczeństwa?
   e. Co to jest za schemat RSA-PSS? Dlaczego zaleca się jego używanie zamiast schematu RSA PKCS#1.5?
   f. Po co w formacie klucza prywatnego ANSI zachowuje się wartości $p$ i $q$?
   g. Jakie są inne ataki na schemat RSA oprócz tych opisanych w pkt. 2?
   h. Dlaczego moduł $N$ nie może być używany więcej niż raz?