Pierwsza Ławka · zestaw działowy · liceum i technikum

Dział 13

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny

Szyfrowanie symetryczne i asymetryczne, RSA na małych liczbach, funkcje skrótu, podpis elektroniczny i certyfikaty, uwierzytelnianie wieloskładnikowe i przechowywanie haseł. Osiem kart: teoria z przykładami, ćwiczenia, utrwalenie w formule CKE, zadanie przy komputerze z plikami danych (ZIP), kartkówka i sprawdzian w grupach A i B – z kluczem i zasadami oceniania.

Informatyka | klasa 1 LO i technikum | Dział 13 | zakres rozszerzony (ZR)

	Karta	Rodzaj	Poziom	Zadania
1	**Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – najważniejsze wiadomości** Teoria, przykłady krok po kroku, rozgrzewka	Najważniejsze wiadomości	ZR	3 zad. · 6 pkt
2	**Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – ćwiczenia podstawowe** Ćwiczę umiejętności z działu	Ćwiczenia	ZR	6 zad. · 15 pkt
3	**Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – utrwalenie** Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych	Zadania typu maturalnego	ZR	6 zad. · 10 pkt
4	**Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – w praktyce, przy komputerze** Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt	Ćwiczenia	ZR	2 zad. · 8 pkt
5	**Kartkówka: RSA, podpis i uwierzytelnianie – grupa A** 15 minut · 7 pkt · bez komputera	Kartkówka	ZR	4 zad. · 7 pkt 15 min
6	**Kartkówka: RSA, podpis i uwierzytelnianie – grupa B** 15 minut · 7 pkt · bez komputera	Kartkówka	ZR	4 zad. · 7 pkt 15 min
7	**Sprawdzian: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – grupa A** 45 minut · 22 pkt · bez komputera	Sprawdzian	ZR	10 zad. · 22 pkt 45 min
8	**Sprawdzian: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – grupa B** 45 minut · 22 pkt · bez komputera	Sprawdzian	ZR	10 zad. · 22 pkt 45 min

PIERWSZA ŁAWKA

Jak korzystać z zestawu

- Karta 1 przypomina najważniejsze wiadomości i pokazuje przykłady rozwiązane krok po kroku (z kodem w Pythonie albo opisem kroków w aplikacji) – od niej warto zacząć.
- Karta 2 to ćwiczenia podstawowe, karta 3 – utrwalenie w formule zadań maturalnych CKE (ramka „Jak to sprawdza CKE").
- Karta 4 to praca przy komputerze: pliki danych są w archiwum ZIP dołączonym do zestawu, a odpowiedzi kontrolne dla plików *_przyklad (pliki o nazwie z dopiskiem „_przyklad") podano przy zadaniach. Wyniki zapisuje się w plikach wynikiN.txt – jak na maturze.
- Karty 5–6 to kartkówka (15 minut), karty 7–8 – sprawdzian (45 minut), obie w grupach A i B: te same typy zadań i ta sama punktacja, inne dane. Kartkówka i sprawdzian nie wymagają komputera.
- Klucz zawiera odpowiedzi, przykładowe programy, zasady oceniania w stylu CKE i numery wymagań podstawy. Osobny plik „Powtórka" zbiera teorię, typowe błędy i test „Sprawdź się".

Oznaczenia

ZP	zakres podstawowy
ZR	zakres rozszerzony (treści tylko dla ZR)
(0–2 pkt)	liczba punktów możliwych do zdobycia za zadanie
Zadanie 4.1.	wiązka zadań do wspólnego materiału źródłowego
Jak to sprawdza CKE	typ zadania i typowy błąd ze sprawozdań CKE
Na maturze	wymagania z podstawy programowej, których dotyczy temat

Wymagania z podstawy programowej (Dz.U. 2024 poz. 1019)

- **V.R1** objaśnia rolę technik uwierzytelniania, kryptografii i podpisu elektronicznego w ochronie i dostępie do informacji;
- **V.R2** omawia znaczenie algorytmów szyfrowania i składania podpisu elektronicznego.
- **I+II.R3f** objaśnia oraz porównuje podstawowe metody i techniki algorytmiczne oraz struktury danych, wykorzystując przykłady problemów i algorytmów, w szczególności: f) metodę szyfrowania z kluczem publicznym i jej zastosowanie w podpisie elektronicznym,

Materiał oryginalny – żadne zadanie nie jest kopią zadania CKE ani podręcznika; dane w plikach są fikcyjne. Podstawa programowa LO i technikum w brzmieniu z 2024 r. (Dz.U. 2024 poz. 1019). Wymagania: V.R1, V.R2, I+II.R3f. Miejsce w programach nauczania: „Informatyka na czasie" (Nowa Era) ZR – część 1 (nazwy serii podano tylko informacyjnie). Do użytku własnego i pracy z własnymi uczniami.

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – najważniejsze wiadomości

Teoria, przykłady krok po kroku, rozgrzewka

ZR </> KARTA 1/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 6 pkt

Najważniejsze wiadomości · 6 pkt

Przypomnij sobie

- **Szyfrowanie symetryczne**: ten sam tajny klucz szyfruje i odszyfrowuje (np. AES). Szybkie, ale obie strony muszą wcześniej bezpiecznie wymienić klucz.
- **Szyfrowanie asymetryczne (z kluczem publicznym)**: każda osoba ma parę kluczy – **publiczny** (jawny) i **prywatny** (tajny). Wiadomość do osoby B szyfrujemy kluczem **publicznym B**; odszyfrować ją może tylko B swoim kluczem prywatnym. Przykład: RSA.
- **RSA w małych liczbach**: wybieramy liczby pierwsze p, q; $n = p \cdot q$, $\varphi = (p - 1)(q - 1)$; e względnie pierwsze z φ; d takie, że $1 < d < \varphi$ i $e \cdot d \bmod \varphi = 1$. Klucz publiczny (n, e), prywatny (n, d). Szyfrowanie $c = m^e \bmod n$, odszyfrowanie $m = c^d \bmod n$. Bezpieczeństwo opiera się na trudności rozkładu dużego n (2048 bitów i więcej) na czynniki.
- **Funkcja skrótu** (np. SHA-256 – skrót 256-bitowy) zamienia dowolne dane na krótki „odcisk”; mała zmiana danych daje zupełnie inny skrót, a z samego skrótu nie da się odtworzyć danych. Służy do kontroli integralności i przechowywania haseł.
- **Podpis elektroniczny**: nadawca liczy skrót h dokumentu i „szyfruje” go swoim **kluczem prywatnym**: $s = h^d \bmod n$. Odbiorca liczy $s^e \bmod n$ **kluczem publicznym nadawcy** i porównuje z własnym skrótem. Zgodność = dokument nie został zmieniony (integralność) i podpisał go właściciel klucza (autentyczność, niezaprzeczalność).
- **Certyfikat** wiąże klucz publiczny z tożsamością właściciela i jest podpisany przez urząd certyfikacji (CA). Tak przeglądarka sprawdza serwer w HTTPS (TLS). W UE **kwalifikowany podpis elektroniczny** ma skutek prawny równoważny podpisowi własnoręcznemu (rozporządzenie eIDAS, art. 25 ust. 2). W Polsce działają też podpis zaufany i podpis osobisty (e-dowód).
- **Uwierzytelnianie** – potwierdzenie tożsamości. Czynniki: coś, co **wiesz** (hasło, PIN), coś, co **masz** (telefon z aplikacją, klucz sprzętowy), coś, czym **jesteś** (odcisk palca, twarz). **Uwierzytelnianie wieloskładnikowe** (2FA/MFA) łączy co najmniej dwa różne czynniki.
- **Hasła w serwisie** przechowuje się nie jawnie, lecz jako skrót z losową **solą** (specjalne, celowo wolne funkcje, np. bcrypt, Argon2) – wyciek bazy nie ujawnia haseł wprost, a sól utrudnia użycie gotowych tablic skrótów.
- **Szybkie liczenie potęg modulo**: w każdym kroku bierzemy resztę – $(a \cdot b) \bmod n = ((a \bmod n) \cdot (b \bmod n)) \bmod n$, więc liczby nie rosną. W Pythonie: pow(m, e, n).

Przykłady z rozwiązaniem

Przykład 1. Utwórz klucze RSA dla p = 5, q = 11, e = 3. Zaszyfruj m = 7 i odszyfruj wynik.

n = 55, φ = 4 · 10 = 40. NWD(3, 40) = 1. d: 3 · 27 = 81 = 2 · 40 + 1 → d = 27. Klucz publiczny (55, 3), prywatny (55, 27).

$c = 7^3 \bmod 55 = 343 \bmod 55 = 13$. $m = 13^{27} \bmod 55 = 7$.

```
n, e, d = 55, 3, 27
c = pow(7, e, n)
print(c, pow(c, d, n))      # 13 7
```

Przykład 2. Anna podpisuje skrót h = 20 kluczem prywatnym (n = 77, d = 43); Bartek ma klucz publiczny Anny (77, 7). Sprawdź podpis.

$s = 20^{43} \bmod 77 = 69$. Bartek liczy $69^7 \bmod 77 = 20 = h$ – podpis jest poprawny. Gdyby ktoś zmienił dokument, skrót byłby inny i porównanie by się nie zgodziło.

```
def potega_mod(a, b, n):      # a do potęgi b modulo n
    w = 1
    for i in range(b):
        w = (w * a) % n
    return w

s = potega_mod(20, 43, 77)
print(s, potega_mod(s, 7, 77))
```

Przykład 3. Dlaczego serwis nie powinien przechowywać haseł jawnie ani jako zwykły skrót bez soli?

Jawne hasła po wycieku bazy są od razu znane. Zwykły skrót (bez soli) dla popularnych haseł jest taki sam u wszystkich użytkowników i można go znaleźć w gotowych tablicach. Losowa sól sprawia, że takie same hasła mają różne skróty, a wolna funkcja (bcrypt, Argon2) spowalnia zgadywanie.

```
import hashlib, os
sol = os.urandom(16)
skrot = hashlib.pbkdf2_hmac("sha256", b"haslo", sol, 200_000)
print(len(skrot) * 8)        # 256 bitów
```

Zadania

Zadanie 1. (0–1 pkt) ..... / 1

Aby wysłać Ewie zaszyfrowaną wiadomość, którą tylko ona odczyta, trzeba użyć

A. klucza prywatnego Ewy B. własnego klucza publicznego C. własnego klucza prywatnego D. klucza publicznego Ewy

Zadanie 2. (0–2 pkt) ..... / 2

Uzupełnij (RSA, p = 3, q = 11).

1. n = _____

2. φ = (p − 1)(q − 1) = _____

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – ćwiczenia podstawowe

Ćwiczę umiejętności z działu

ZR </> KARTA 2/8

Imię i nazwisko: Klasa: Data: Wynik: ..... / 15 pkt

Ćwiczenia · 15 pkt

Zadanie 1. (0–3 pkt) ..... / 3

Utwórz klucze RSA dla p = 3, q = 11, e = 3. Uzupełnij tabelę.

n	ϕ	d

Zadanie 2. (0–2 pkt) ..... / 2

Klucz publiczny (33, 3), prywatny (33, 7).

1. zaszyfruj m = 4: c = _____
2. odszyfruj c = 31: m = _____

Zadanie 3. (0–1 pkt) ..... / 1

Uzupełnij funkcję potega_mod(a, b, n).

```
def potega_mod(a, b, n):      # a do potęgi b modulo n
    w = 1
    for i in range(b):
        w = luka
    return w
```

1. luka: _____

Zadanie 4. (0–4 pkt) ..... / 4

Połącz cel z kluczem, którego się używa.

1. zaszyfrowanie listu do Piotra
2. złożenie podpisu przez Olę
3. sprawdzenie podpisu Oli
4. odczytanie listu przez Piotra

a) klucz publiczny Piotra
b) klucz prywatny Piotra
c) klucz prywatny Oli
d) klucz publiczny Oli

1 ____ 2 ____ 3 ____ 4 ____

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – utrwalenie

Zadania w formule CKE: polecenia i punktacja jak w arkuszach maturalnych

ZR </> KARTA 3/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 10 pkt

Zadania typu maturalnego · 10 pkt

Jak to sprawdza CKE

Kryptografię z kluczem publicznym CKE sprawdza zadaniami P/F: 2023 zad. 4 – do czego służy klucz prywatny w szyfrowaniu asymetrycznym (1 pkt za komplet). Bezpieczeństwo pojawia się też w „Dokończ zdanie" (2025 zad. 4 – rodzaj szkodliwego programu, 1 pkt). Zapamiętaj: szyfruje się kluczem publicznym odbiorcy, podpisuje kluczem prywatnym nadawcy.

Zadanie 1. (0–1 pkt) ..... / 1

Oceń prawdziwość podanych zdań. Zaznacz P, jeśli zdanie jest prawdziwe, albo F – jeśli jest fałszywe.

1. W podpisie elektronicznym nadawca używa swojego klucza prywatnego.	P	F
2. Znając klucz publiczny, można łatwo obliczyć klucz prywatny dla n o długości 2048 bitów.	P	F
3. Zmiana jednego znaku dokumentu po podpisaniu zostanie wykryta przy weryfikacji.	P	F

Zadanie 2. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Funkcja skrótu stosowana w podpisie elektronicznym

A. daje skrót o długości zależnej od długości dokumentu

B. pozwala odtworzyć dokument ze skrótu

C. dla różnych danych praktycznie zawsze daje różne skróty

D. wymaga klucza publicznego odbiorcy

Zadanie 3. (0–1 pkt) ..... / 1

Dokończ zdanie. Zaznacz właściwą odpowiedź spośród podanych.
Dla RSA z $p = 5$, $q = 7$ i $e = 5$ kluczem prywatnym d $(1 < d < \varphi)$ jest

A. 7 B. 5 C. 19 D. 11

Zadanie 4. (0–2 pkt) ..... / 2

Uzupełnij tabelę – wpisz, czyj klucz (nadawcy / odbiorcy) i jaki (publiczny / prywatny) jest używany.

czynność	klucz (czyj i jaki)
szyfrowanie wiadomości	
odszyfrowanie wiadomości	
złożenie podpisu	
weryfikacja podpisu	

Zadanie 5. (0–2 pkt) ..... / 2

Klucz publiczny Kuby to (n, e) = (65, 5). Dostałeś dokument ze skrótem h = 22 i podpisem s = 42. Sprawdź, czy podpis jest poprawny.

Miejsce na odpowiedź

Zadanie 6. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla danych e i φ znajdzie najmniejsze d > 0, dla którego e · d mod φ = 1, albo zwróci 0, gdy takie d nie istnieje.

Przykład: dla e = 3 i φ = 40 wynikiem jest 27.

Uwaga: Twój algorytm może używać tylko zmiennych przechowujących pojedyncze liczby całkowite, operacji arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie całkowite i reszta z dzielenia), porównań, operatorów logicznych, instrukcji warunkowych, pętli, przypisań i własnych funkcji. Nie wolno używać funkcji wbudowanych, tablic ani list ani zamieniać liczby na napis.

Specyfikacja

Dane: e, φ – liczby całkowite, $1 < e < \varphi \leq 10^6$

Wynik: d – najmniejsza liczba $1 \leq d < \varphi$, dla której e · d mod φ = 1, albo 0

Miejsce na odpowiedź

PIERWSZA ŁAWKA

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – w praktyce, przy komputerze

Pliki danych w archiwum ZIP · wyniki w plikach wynikiN.txt

ZR </> KARTA 4/8

Imię i nazwisko: ..... Klasa: ..... Data: ..... Wynik: ..... / 8 pkt

Ćwiczenia · 8 pkt

Zadanie 1. (0–5 pkt)

Klucze RSA

Opis danych

Plik **klucze.txt** zawiera 300 wierszy; w każdym trzy liczby p, q, e (p i q – różne liczby pierwsze, 1 < e < φ) oddzielone spacją. Klucz jest poprawny, gdy NWD(e, φ) = 1, gdzie φ = (p − 1)(q − 1). Pierwsze wiersze:

```
367 13 51
83 197 884
19 13 59
```

Plik **klucze_przyklad.txt** zawiera 10 wierszy w tym samym formacie.

Do dyspozycji masz plik **klucze.txt** oraz mniejszy plik **klucze_przyklad.txt**. Odpowiedzi dla danych przykładowych podano przy zadaniach – najpierw sprawdź na nich swój program.

Napisz program (lub kilka programów), który(-e) znajdzie(-ą) i da(dzą) odpowiedzi do podanych zadań. Odpowiedzi zapisz w pliku **wyniki1.txt**, a każdą z nich poprzedź numerem odpowiedniego zadania.

Do oceny oddajesz: plik tekstowy **wyniki1.txt** z odpowiedziami do zadań 1.1–1.2 (każda poprzedzona numerem zadania) oraz pliki z kodem źródłowym programów, np. **zadanie1.py**.

Dane fikcyjne, przygotowane do zadania.

Zadanie 1.1. (0–2 pkt) ..... / 2

Podaj, ile kluczy jest poprawnych. *Dla danych z pliku przykładowego odpowiedź: 3.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 1.2. (0–3 pkt) ..... / 3

Dla poprawnych kluczy oblicz klucz prywatny d (1 ≤ d < φ, e · d mod φ = 1). Podaj n = p · q i d dla klucza z największym d, oddzielone spacją. *Dla danych z pliku przykładowego odpowiedź: 105187 82153.*

Odpowiedź zapisz w pliku wyniki1.txt

Zadanie 10. (0–3 pkt) ..... / 3

Zapisz w pseudokodzie lub w wybranym języku programowania algorytm, który dla liczb a i n znajdzie najmniejsze $k \geq 1$, dla którego $a^k \bmod n = 1$, nie obliczając potęg w całości.

Przykład: dla a = 6, n = 65 wynikiem jest 12.

Uwaga: Twój algorytm może używać tylko zmiennych przechowujących pojedyncze liczby całkowite, operacji arytmetycznych (dodawanie, odejmowanie, mnożenie, dzielenie całkowite i reszta z dzielenia), porównań, operatorów logicznych, instrukcji warunkowych, pętli, przypisań i własnych funkcji. Nie wolno używać operatora potęgowania ani funkcji wbudowanych, tablic ani list ani zamieniać liczby na napis.

Specyfikacja

Dane: a, n – liczby całkowite, $2 \leq n \leq 10^6$, $1 \leq a < n$, NWD(a, n) = 1

Wynik: k – najmniejsza liczba $k \geq 1$, dla której $a^k \bmod n = 1$

Miejsce na odpowiedź

PIERWSZA ŁAWKA

CO JESZCZE JEST W ŚRODKU

Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – kartkówka, sprawdzian i karty pracy z powtórką, informatyka, klasa 1 LO i...

Podgląd pokazał 9 z 28 stron. Pełna wersja: 28 stron A4, PDF do druku + DOCX do edycji.

- Karta 1: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – najważniejsze wiadomości – 3 zadania, 6 pkt
- Karta 2: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – ćwiczenia podstawowe – 6 zadań, 15 pkt
- Karta 3: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – utrwalenie – 6 zadań, 10 pkt
- Karta 4: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – w praktyce, przy komputerze – 2 zadania, 8 pkt
- Karta 5: Kartkówka: RSA, podpis i uwierzytelnianie – grupa A – 4 zadania, 7 pkt
- Karta 6: Kartkówka: RSA, podpis i uwierzytelnianie – grupa B – 4 zadania, 7 pkt
- Karta 7: Sprawdzian: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – grupa A – 10 zadań, 22 pkt
- Karta 8: Sprawdzian: Kryptografia: klucz publiczny, uwierzytelnianie i podpis elektroniczny – grupa B – 10 zadań, 22 pkt Karta 1 przypomina teorię i pokazuje przykłady rozwiązane k

Kup pełną wersję – plik od razu na e-mail

pierwszalawka.pl/material/informatyka-liceum-klasa-1-zr-dzial-13-kryptografia-klucz-publiczny-uwierzytelnianie