

Kryptoanaliza stosowana 2025

Lista zadań nr 5: proste ataki na RSA

Na zajęcia 17 listopada 2025

Zadanie 1 (2 pkt). Ułamki łańcuchowe (*continued fractions*) pozwalają m. in. na znajdowanie ułamków zwykłych o niewielkim mianowniku, które bardzo dobrze aproksymują liczby rzeczywiste. Niech będzie dana liczba rzeczywista $x > 0$. Definiujemy:

$$\begin{aligned} x_0 &= x, \\ x_{i+1} &= \frac{1}{x_i - a_i}, \quad \text{dla } i \in \mathbb{N}, \\ a_i &= \lfloor x_i \rfloor, \quad \text{dla } i \in \mathbb{N}, \\ p_{-2} &= 0, \\ q_{-2} &= 1, \\ p_{-1} &= 1, \\ q_{-1} &= 0, \\ p_i &= a_i p_{i-1} + p_{i-2}, \quad \text{dla } i \in \mathbb{N}, \\ q_i &= a_i q_{i-1} + q_{i-2}, \quad \text{dla } i \in \mathbb{N}. \end{aligned}$$

Dla $x \in \mathbb{Q}$ obliczenie kończy się dla pewnego $i \in \mathbb{N}$, gdy $x_i = a_i$ i wówczas $x = p_i/q_i$. Dla $x \notin \mathbb{Q}$ ciąg jest nieskończony oraz

$$\lim_{i \rightarrow \infty} \frac{p_i}{q_i} = x.$$

Ponadto

$$\frac{p_i}{q_i} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\dots + a_i}}} \stackrel{\text{ozn.}}{=} [a_0; a_1, a_2, \dots, a_i].$$

Na przykład dla $x = \pi$ mamy:

i	a_i	p_i/q_i	$\pi - p_i/q_i$
0	3	3/1	$1.41 \cdot 10^{-1}$
1	7	22/7	$-1.26 \cdot 10^{-3}$
2	15	333/106	$8.32 \cdot 10^{-5}$
3	1	355/113	$-2.66 \cdot 10^{-7}$
4	292	103993/33102	$5.77 \cdot 10^{-10}$
5	1	104348/33215	$-3.31 \cdot 10^{-10}$
6	1	208341/66317	$1.22 \cdot 10^{-10}$
7	1	312689/99532	$-2.91 \cdot 10^{-11}$
8	2	833719/265381	$8.71 \cdot 10^{-12}$
9	1	1146408/364913	$-1.61 \cdot 10^{-12}$
10	3	4272943/1360120	$4.04 \cdot 10^{-13}$
11	1	5419351/1725033	$-2.21 \cdot 10^{-14}$
12	14	80143857/25510582	$5.79 \cdot 10^{-16}$
13	2	165707065/52746197	$-1.64 \cdot 10^{-16}$
14	1	245850922/78256779	$7.81 \cdot 10^{-17}$
15	1	411557987/131002976	$-1.93 \cdot 10^{-17}$
16	2	1068966896/340262731	$3.07 \cdot 10^{-18}$
17	2	2549491779/811528438	$-5.51 \cdot 10^{-19}$
18	2	6167950454/1963319607	$7.62 \cdot 10^{-20}$
19	2	14885392687/4738167652	$-3.12 \cdot 10^{-20}$
20	1	21053343141/6701487259	$2.61 \cdot 10^{-22}$
21	84	1783366216531/567663097408	$-1.22 \cdot 10^{-24}$
22	2	3587785776203/1142027682075	$3.14 \cdot 10^{-25}$
23	1	5371151992734/1709690779483	$-1.97 \cdot 10^{-25}$
24	1	8958937768937/2851718461558	$7.72 \cdot 10^{-27}$

W powyższej tabeli $p_1/q_1 = 22/7$ jest słynnym oszacowaniem π , używanym już przez starożytnych Greków, zaś

$$\pi = [3; 7, 15, 1, 292, 1, 1, 1, 2, 1, 3, 1, 14, 2, 1, 1, 2, 2, 2, 1, 84, 2, 1, 1, 15, 3, 13, 1, 4, 2, 6, 6, 99, 1, 2, 2, 6, 3, 5, 1, \\ 1, 6, 8, 1, 7, 1, 2, 3, 7, 1, 2, 1, 1, 12, 1, 1, 1, 3, 1, 1, 8, 1, 1, 2, 1, 6, 1, 1, 5, 2, 2, 3, 1, 2, 4, 16, 1, 161, 45, 1, \dots]$$

jest rozwinięciem π w ułamek łańcuchowy.¹

Mamy też ciekawe twierdzenie: jeżeli dla pewnych liczb naturalnych $r, s \in \mathbb{N}$ jest $|x - \frac{r}{s}| < \frac{1}{2s^2}$, to istnieje takie $i \in \mathbb{N}$, że $\frac{r}{s} = \frac{p_i}{q_i}$.

Rozważmy kryptosystem RSA z kluczem jawnym (n, e) i tajnym (n, d) , gdzie $n = pq$ i $ed \equiv 1 \pmod{\phi(n)}$. Oczywiście $1 \leq e, d < \phi(n)$. Przypuśćmy że $q < p < 2q$ oraz że $d < \frac{1}{3}\sqrt[4]{n}$. Mamy $q^2 < n$ (bo $qq < pq = n$). Ponieważ $p < 2q$, to

$$n - \phi(n) = pq - (p-1)(q-1) = p + q - 1 < 3q < 3\sqrt{n}.$$

Skoro $ed \equiv 1 \pmod{\phi(n)}$, to istnieje takie $k \in \mathbb{N}$, że $ed = 1 + k\phi(n)$. Ponieważ $e < \phi(n)$, to

$$k\phi(n) = ed - 1 < ed < \phi(n)d < \phi(n)\frac{1}{3}\sqrt[4]{n}.$$

Zatem $k < \frac{1}{3}\sqrt[4]{n}$. Stąd

$$kn - ed = kn - 1 - k\phi(n) = k(n - \phi(n)) - 1 < k(n - \phi(n)) < \left(\frac{1}{3}\sqrt[4]{n}\right)(3\sqrt{n}) = n^{3/4}.$$

Ponieważ $k(n - \phi(n)) - 1 > 0$, to także $kn - ed > 0$. Stąd

$$0 < \frac{k}{d} - \frac{e}{n} < \frac{1}{d\sqrt[4]{n}} < \frac{1}{3d^2} < \frac{1}{2d^2}.$$

Ułamek k/d jest więc na mocy cytowanego wyżej twierdzenia równy pewnemu rozwinięciu łańcuchowemu p_i/q_i ułamka e/n . Ponieważ e/n jest wymierne, to ciąg p_i/q_i jest skończony (i ma długość logarytmiczną względem n).

Jak sprawdzić, że p_i/q_i to k/d ? Skoro $ed = 1 + k\phi(n)$, to $\phi(n) = \frac{ed-1}{k}$. Obliczymy zatem $c_i = \frac{eq_i-1}{p_i}$. Jeśli c_i nie jest całkowite, to na pewno $p_i/q_i \neq k/d$. W przeciwnym razie rozważmy wielomian

$$(x-p)(x-q) = x^2 - (p+q)x + n = x^2 - (n - \phi(n) + 1)x + n.$$

Jego pierwiastkami są liczby p i q . Jeśli więc $c_i = \phi(n)$, to pierwiastkami wielomianu

$$x^2 - (n - c_i + 1)x + n$$

są liczby p i q . Obliczamy zatem te pierwiastki i sprawdzamy, czy są większymi niż jeden liczbami naturalnymi oraz ich iloczynem jest n . Zauważmy, że jeśli założenia o d , p i q są spełnione, to na pewno natrafimy w ciągu p_i/q_i na odpowiedni wyraz. W przeciwnym razie procedura *może* (ale nie *musi*) zakończyć się niepowodzeniem. Zauważmy ponadto, że $k/d > e/n$, zaś kolejne aproksymacje p_i/q_i ułamka e/n są na przemian większe i mniejsze od e/n . Zatem można testować co drugą wartość p_i/q_i . Zauważmy w końcu, że d jest nieparzyste (bo względnie pierwsze z $p-1$), więc jeśli q_i jest parzyste, to możemy od razu odrzucić ten ułamek.

Na przykład dla $n = 1966981193543797$ oraz $e = 323815174542919$ mamy:

i	p_i/q_i	$p_i/q_i - e/n$	c_i
0	0	-323815174542919 / 1966981193543797	—
1	1/6	24090146286283 / 11801887161262782	1942891047257513 / 1
2	13/79	-10643272821240 / 155391514289959963	25581398788890600 / 13
3	27/164	2803600643803 / 322584915741128708	53105688625038715 / 27
4	94/571	-2232470889831 / 112314626151350807	92449232332003374 / 47
5	121/735	571129753972 / 1445731177254690795	238004153289045464 / 121
6	457/2776	-519081627915 / 5460339793277580472	898910924531143143 / 457
7	578/3511	52048126057 / 6906070970532271267	1966981103495136 / 1
8	5659/34375	-50648493402 / 67614978528068021875	11131146624912840624 / 5659
9	6237/37886	1399632655 / 74521049498600293142	12268061702733029233 / 6237
10	230191/1398271	-261717822 / 2750372760477678574987	452781367923301893048 / 230191
11	1157192/7029241	91043545 / 13826384851886993168077	113807450659621247239 / 578596
12	2544575 / 15456753	-79630732 / 30403142464251664911141	5005131170561786882006 / 2544575
13	3701767 / 22485994	11412813 / 44229527316138658079218	428312121875354669205 / 217751
14	2475517 / 150327217	-11153854 / 295780306361083613386449	48692967601847963140922 / 24755177
15	28456944 / 172858711	258959 / 34000983367722271465667	1166130701536020677446 / 592853
16	1248403769 / 7583297290	-18617 / 14916203154481641286410130	245558735572194641389509 / 1248403769
17	16257705941 / 98755723481	16938 / 194250650841938558994797357	31978601836112259330581038 / 16257705941
18	17506109710 / 106339020771	-1679 / 209166853996420200281207487	17217094285842226985985274 / 8753054855
19	191318803041 / 1162145931191	148 / 228591910806140561806872227	376320487552956799050286528 / 191318803041
20	2122012943161 / 12889944263872	-51 / 25354277952863966380156801984	4173959551654209243525122367 / 2122012943161
21	4435344689363 / 26942034458935	46 / 52994475096534073322120476195	8724239590861375286100531264 / 4435344689363
22	6557357632524 / 39831978722807	-5 / 78348753049398039702277278179	3224549785628896132406413408 / 1639339408131
23	63451563382079 / 385429842964198	1 / 758133254116430642615979806	124808031873501636052731413961 / 63451563382079
24	323815174542919 / 1966981193543797	0	636938358510023764793282723442 / 323815174542919

¹Skrypty w bc(1) wyznaczające dowolnie długie rozwinięcia łańcuchowe są dołączone do niniejszej listy jako załączniki.

Ponieważ $\lfloor \frac{1}{3} \sqrt[4]{n} \rfloor = 2219$, to możemy mieć pewność, że procedura się powiedzie, jeśli $d \leq 2219$. Tylko dla dwóch wartości i liczba c_i jest całkowita. Ponieważ $q_1 = 6$ jest parzyste, nie może być kandydatem na d i $i = 1$ odrzucamy. Pozostaje $i = 7$. Wyróżnik równania kwadratowego

$$x^2 - 90048662x + 1966981193543797 = 0$$

to $\Delta = 240836753815056$. Ponieważ $\sqrt{\Delta} = 15518916$, to

$$x_1 = 37264873, \quad x_2 = 52783789.$$

Istotnie $x_1 \cdot x_2 = n$, zatem $p = 52783789$, $q = 37264873$, $\phi(n) = 1966981103495136$, $d = 3511$. Rzeczywiście $de \equiv 1 \pmod{\phi(n)}$. Zauważmy przy tym, że $d > 2219$.

Zaimplementuj powyższą procedurę.² W Pythonie przyda się moduł `fractions`.

Zadanie 2 (1 pkt). Oto szyfrogram RSA:

$$\begin{aligned} n &= 13068749442931688059258760359152138392041262881104930893575297281461774701031509597192 \\ &\quad 31458770646881379929026855960972167209968548539192187271593900212271487338557623868585 \\ &\quad 55333362302124683761104846232605563413435009531762821185586791939352214160535318857072 \\ &\quad 814374111130766141753918111259889897345003064525351 \\ e &= 28693214582397890025989589693388201307112484632946573521392115165335417643770107378187 \\ &\quad 4024575507869521635631462613131150296602028075822980873282558889275062528369798449946 \\ &\quad 87628559565870260463698558337208471923781003056739711200134104237950290253050911271418 \\ &\quad 27911114988375014692107707438719088975457125194099 \\ c &= 55454605216274283003615025402579985926872868938531978788960945200071288606272984986984 \\ &\quad 69300794109301794822842657665904435611599310147533430695328576993235090275831943192242 \\ &\quad 08925961396839293385623204978893214454356327617481509433312399978959797478604040508454 \\ &\quad 49692343927454617961984814625948913598527195055849 \end{aligned}$$

Coś poszło źle, prawda? Kryptografia wymaga skupienia, a ja się nie mogłem oderwać od Stephena Kinga...

Zadanie 3 (2 pkt). Dla $i = 0, 1, \dots$ generuj losowo (np. po 1024) takie $p \in [2^{512}, 2^{513}]$ i $q \in [2^{512}, 2^{513}]$ oraz $d \in [2^{253+i}, 2^{254+i}]$, że $d \perp \phi(n)$ i sprawdź, czy dają się zaatakować procedurą opisaną w zadaniu 1. Dla $i = 0$ mamy $(3d)^4 < (3 \cdot 2^{254})^4 = 81 \cdot 2^{1016} < 2^7 \cdot 2^{1016} = 2^{1023}$, a skoro $p, q \geq 2^{512}$, to $n \geq 2^{1024}$, więc $d < \frac{1}{3} \sqrt[4]{n}$ i każde d powinno dać się zaatakować. Dla większych i — nie koniecznie. Narysuj histogram, w którym na osi poziomej jest i , zaś na pionowej — $\log_2(P_i)$, gdzie P_i jest odsetkiem wylosowanych d , które udało się skutecznie zaatakować. Czy w praktyce opisany w zadaniu 1 atak jest więc zagrożeniem?

Zadanie 4 (1 pkt). Zaimplementuj OAEP zgodnie ze standardem *PKCS #1: RSA Cryptography Specifications Version 2.1* (RFC 3447, Feb. 2003). Przyjmij, że etykieta L jest pustym ciągiem.

Zadanie 5 (1 pkt). Oto szyfrogram RSA, w którym szyfrowana wiadomość została przygotowana zgodnie ze standardem OAEP przy użyciu funkcji mieszącej SHA256. Odszyfruj wiadomość.

$$\begin{aligned} n &= 13297651856842887359099069899047388814743418668405579064754225771118350430079356789476 \\ &\quad 32581234945582640116088299840424952708947139387367462896131250529790259360044676484887 \\ &\quad 90223063947335152106970185641201512558554026954048393654332511691849974991216693754764 \\ &\quad 137851682541757604893914324078124535978192415120291 \\ e &= 65537 \\ d &= 90691539900408400794917546924290269291024396023883328113480877869245908291996431087663 \\ &\quad 96613128211957713241191805845167479627051938477068448215264431531810257139719601048137 \\ &\quad 03956952263236811369026284946941171409160576085747594533912551737788184963219401363149 \\ &\quad 11468520140003952162724505217290328347539832915473 \\ c &= 32285987280270223193394300987555362945882719223146559544577770972899088517077123906704 \\ &\quad 03496161375270809875472814027194165823568855250235352925887514844611478292228044277487 \\ &\quad 19774725797712776109678306328118773548564613745938098296972661484083964416035673287905 \\ &\quad 50104677162545537524254825671420002401958272746803 \end{aligned}$$

²Atak został wynaleziony przez Michaela Wienera i opisany w pracy: Cryptanalysis of short RSA secret exponents, *IEEE Trans. Information Theory*, **36**:553–558(1990). Przestępny opis znajduje się w: Wade Trappe, Lawrence C. Washington, *Introduction to Cryptography with Coding Theory*, 2nd ed., Pearson 2006, podrozdział 6.2.1, str. 170–172, skąd też zaczerpnąłem podany w zadaniu przykład. Zob. także: Douglas R. Stinson, Maura B. Paterson, *Cryptography. Theory and Practice*, 4th ed., CRC Press, 2019, podrozdział 6.7.3, str. 228–232.

Zadanie 6 (1 pkt). Oto szyfrogram RSA, w którym szyfrowana wiadomość została przygotowana zgodnie ze standardem OAEP przy użyciu funkcji mieszającej SHA256. Złam ten szyfrogram!

$$\begin{aligned} n &= 71502864762146481719921706717489647274401371264339912861896668788523408898399987584458 \\ &\quad 8674271503779828631828277232104969492966706566918653613986624948820418524016225055718 \\ &\quad 69102836757569010114971055247817758777399370331067829716957112796906336760590660992917 \\ &\quad 99230305634094313871898669144926153021806765337119 \\ e &= 2021535990211590963017392592316424966597438880474232111927751687589711189711969224963 \\ &\quad 80667821335075997803106339714897610001405009507387105725190113425251075371250635294214 \\ &\quad 68830372847875702143678744838759071318617997104923057075420624171678235157677719289162 \\ &\quad 9109675421410740010665647739359531488881247608893 \\ c &= 60342465832449322744726634842148045802439244500089483738363609568925340232650344873572 \\ &\quad 85251194445070659860435560566120651631991471715812359773527732217946491979140099916033 \\ &\quad 92736775295359354675120054756619048544319079259553772887278092066179912188456929774725 \\ &\quad 10923562573590533382301718414219836335384794307181 \end{aligned}$$

Zadanie 7 (1 pkt). Rozważmy kryptosystem RSA, w którym e jest niewielkie, np. $e = 3$. Trzy osoby: Bob, Chris i Denis mają klucze publiczne $(n_1, 3)$, $(n_2, 3)$, $(n_3, 3)$. Adam wysyła do nich tę samą wiadomość m , wyznaczając $m^3 \bmod n_i$. Jeśli liczby n_1 , n_2 i n_3 są względnie pierwsze, to na mocy Chińskiego Twierdzenia o Resztach Ewa może wyznaczyć $m^3 \bmod n_1 n_2 n_3$. Ponieważ $m^3 < n_1 n_2 n_3$, to Ewa może wyznaczyć m za pomocą zwykłego pierwiastkowania, jak w ataku z małym m . Jeśli n_1 , n_2 i n_3 nie są względnie pierwsze, to Ewa również z łatwością odszyfruje wiadomość, prawda? Przygotuj PoC w którym zademonstrujesz ten atak. Użyj realistycznie dużych kluczy, np. 1024-bitowych.

Zadanie 8 (1 pkt). Przypuśćmy, że Ewa umie skłonić Adama do odszyfrowania podanej mu wiadomości i ujawnienia jej tekstu jawnego (atak *chosen ciphertext*). Bob wysyła do Adama wiadomość $c = m^e \bmod n$. Ewa podshukuje tę wiadomość. Ewa wybiera dowolną wartość $r \in \mathbb{Z}_n^*$ i oblicza $k = cr^e \bmod n$, a następnie wysyła tę wiadomość Adamowi z prośbą o odszyfrowanie. Adam się dziwi, po co jej taki losowy ciąg, ale odsyła jej $k^d \bmod n$. Przygotuj odpowiedni PoC. Przemyśl scenariusze, w których taki atak jest realistyczny.

Zadanie 9 (1 pkt). Aby obniżyć koszty generowania kluczy, pewna firma zainstalowała w swoim produkcie klucze RSA w taki sposób, że *modulus n* został wygenerowany raz, zaś dla każdego klucza wybrano unikatowy wykładnik publiczny e (i obliczono d). Zauważ, że właściciel jednego urządzenia może z łatwością złamać klucz innego (jeśli jest naprawdę właścicielem, tj. ma dostęp do d). To nie jest jeszcze najgorsze! Założymy, że Ewa ma szyfrogramy $c_i = m^{e_i} \bmod n$ dla $i = 1, 2$. Ma też klucze publiczne (n, e_i) . Możemy przyjąć, że $c_1 \perp n$ (w przeciwnym razie złamaliśmy szyfr, prawda?). Możemy zatem obliczyć $c_1^{-1} \bmod n$. Przypuśćmy, że $e_1 \perp e_2$. Za pomocą algorytmu Euklidesa znajdziemy takie r i s , że $re_1 + se_2 = 1$. Jedna z tych liczb jest ujemna, przyjmijmy, że r . Obliczmy $(c_1^{-1})^{-r} c_2^s \bmod n$. Dopracuj szczegóły ataku i przygotuj PoC.

Zadanie 10 (1 pkt). Wiadomość podpisana przez Adama za pomocą RSA to para (m, σ) , gdzie tym razem do szyfrowania Adam używa swojego tajnego klucza: $\sigma = m^d \bmod n$. Bob, chcąc zweryfikować podpis oblicza $\sigma^e \bmod n$. Jeśli tak wyliczona wartość jest równa m , to podpis się zgadza. Mamy tu sporo pułapek:

1. *Egzystencyjne sfałszowanie podpisu.* Ewa wybiera dowolną wartość $\sigma \in \mathbb{Z}_n^*$ i oblicza $m = \sigma^e \bmod n$, gdzie e jest kluczem publicznym Adama, a następnie twierdzi, że (m, σ) to wiadomość podpisana przez Adama.
2. *RSA jest homomorfizmem.* Adam utworzył podpisane wiadomości (m_1, σ_1) i (m_2, σ_2) . Wtedy $(m_1 \cdot m_2 \bmod n, \sigma_1 \cdot \sigma_2 \bmod n)$ oraz $(m^{-1} \bmod n, \sigma^{-1} \bmod n)$ są poprawnie podpisanymi wiadomościami.
3. *Podpisywanie jest operacją odwrotną do szyfrowania.* Bob wysyłał Adama zaszyfrowaną wiadomość $c = m^e \bmod n$. Ewa ją podsłuchała, a następnie prosi Adama o złożenie podpisu na c (tj. obliczenie $\sigma = c^d \bmod n$), np. twierdząc, że c to jej klucz publiczny, który chciałaby uwierzygodnić podpisem Adama. Oczywiście Adam zobaczy, że coś z σ jest nie tak... Dlatego Ewa wybiera losowe $r \in \mathbb{Z}_n^*$ i oblicza $x = r^e c \bmod n$, a następnie prosi Adama o podpisanie x . Obliczona przez Adama $\sigma = x^d \bmod n$ wygląda na losową wartość, ale Ewa z łatwością potrafi z σ wyznaczyć m .

Przygotuj odpowiednie PoC. Rozważ scenariusze, w których powyższe ataki mogłyby znaleźć zastosowanie.³
Ostatni atak łatwo zablokować przyjmując, że *nie wolno używać tych samych kluczy do szyfrowania i uwierzytelniania*. Większość protokołów kryptograficznych przyjmuje tę zasadę.

³ Wsk.: puść wodze wyobraźni!

Zadanie 11 (1 pkt). Niech $E : y^2 = x^3 + ax + b$ będzie krzywą eliptyczną nad ciałem K i niech $L \supseteq K$ będzie rozszerzeniem K . Udowodnij, że $(E(L), +, \infty)$ jest grupą abelową, gdzie działanie $+$ i element ∞ są zdefiniowane jak na wykładzie (żmudne, ale łatwe).

Zadanie 12 (1 pkt). Pokaż, że $E(\mathbb{R})$, gdzie $E : y^2 = x^3 - 3x + 3$, jest topologicznie izomorficzna (homeomorficzna) z okręgiem jednostkowym na płaszczyźnie, tj. zbuduj taką ciągłą bijekcję

$$h : E(\mathbb{R}) \rightarrow \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 = 1\},$$

że h^{-1} też jest ciągła (wcześniej nie żmudne i do tego łatwe).