Kodowanie, Kompresja, Kryptografia

Konspekt VI, 10.V.2022

Kryptografia

$\mathcal{M}$ — zbiór komunikatów, $\mathcal{M} \subset \mathcal{A}^*$
$\mathcal{C}$ — zbiór szyfrogramów, $\mathcal{C} \subset \mathcal{B}^*$
$\mathcal{K}$ — zbiór kluczy szyfrujących

Szyfrowanie przy pomocy klucza $K \in \mathcal{K}$:

$$\mathcal{M} \xrightarrow{s_K} \mathcal{C}$$

Deszyfrowanie: odwzorowanie odwrotne s_K^{-1}.

Przykład: Szyfr Cezara

$$a \in \mathcal{A} \quad \mapsto \quad b = (a+k) \bmod |\mathcal{A}| \in \mathcal{A}$$

a	b	c	....	l	m	n	o	....	x	y	z
			....	y	z	a	b	....			

k

KRYPTOGRAFIA $\mapsto$ XELCGBTENSVN

Szyfr łatwy do złamania na podstawie analizy częstości poszczególnych znaków.

Bardziej skomplikowany szyfr Vigenère'a (XVI w.) polega na użyciu słowa–klucza, którego kolejne litery (odczytywane cyklicznie) wyznaczają przesunięcie alfabetu dla kolejnej szyfrowanej litery. Dla szyfru Cezara jw. kluczem byłoby słowo NNN.

K	R	Y	P	T	O	G	R	A	F	I	A
K	L	U	C	Z	K	L	U	C	Z	K	L
U	C	S	R	S	Y	R	M	C	E	S	L

Ten szyfr również nie jest bezpieczny — zwłaszcza dla krótkiego w porównaniu z szyfrogramem klucza. Dzięki analizie statystycznej i poszukiwaniu podobnych, krótkich ciągów liter można odgadnąć długość klucza. Podział szyfrogramu na odcinki równe długości klucza pozwala rozbić problem na kilka równolegle stosowanych szyfrów Cezara.

Kolejne ulepszenie to szyfr z autokluczem: Pierwsza litera jest ustalona, dalej klucz to sam szyfrowany tekst:

$$\begin{array}{cccccccccccc} K & R & Y & P & T & O & G & R & A & F & I & A \\ M & K & R & Y & P & T & O & G & R & A & F & I \\ W & B & P & N & & & & ... & & & & \end{array}$$

Dłuższy klucz to trudniejsza deszyfracja.

Łamanie szyfru: a) odczytanie pojedynczego szyfrogramu, b) odtworzenie klucza szyfrującego.

Poziomy bezpieczeństwa

1. Zabezpieczenie przez złamaniem szyfru na podstawie znajomości przechwyconej próbki szyfrogramu
2. Zabezpieczenie przez złamaniem szyfru na podstawie znajomości próbki tekstu otwartego i szyfrogramu
3. Zabezpieczenie przez złamaniem szyfru gdy dostępny jest dowolny tekst otwarty i szyfrogram

Omówione przykłady nie gwarantują bezpieczeństwa nawet na poziomie 1 — dostatecznie długa próbka szyfrogramu otwiera możliwości analizy frekwencyjnej. Odczytanie pojedynczego szyfrogramu ujawnia także klucz.

Z drugiej strony pełne zabezpieczenie na poziomie 3 jest niemożliwe, ponieważ alfabety i zbiór kluczy są skończone, więc zawsze można wykonać ekstensywne przeszukanie wszystkich kombinacji. Zabezpieczenie 3 polega na wymuszeniu niepraktycznej złożoności obliczeniowej takiego przeszukania.

Obraz probabilistyczny

M zmienna losowa o wartościach w $\mathcal{M}$, $K \in \mathcal{K}$ zm. losowa niezależna od M.

Szyfrogram jest więc zmienną losową $C = s_K(M) \in \mathcal{C}$. Można myśleć, że M przesyłane jest przez zaszumiony kanał, gdzie szum pochodzi od klucza K.

Definicja: *Wieloznaczność komunikatu* jest to entropia warunkowa $H(M|C)$, natomiast *wieloznaczność klucza* jest to $H(K|C)$.

Lemat: $H(M|C) \leq H(K|C)$.

Dowód.

$$\begin{aligned} H(K|C) &= H(K,C) - H(C) \\ &= H(M,K,C) - H(C) \qquad \text{bo } M = s_K^{-1}(C) \\ &= H(M,K,C) - H(M,C) + H(M,C) - H(C) \\ &= H(K|M,C) + H(M|C) \;\geq\; H(M|C)\,. \end{aligned}$$

Definicja: Szyfr jest *doskonały* jeśli $H(M|C) = H(M)$.

Oznacza to, że $H(M,C) = H(M) + H(C) \quad \Rightarrow$ *niezależność!* (innymi słowy $I(M,C) = 0$.

Lemat: Szyfr doskonały wymaga aby $|\mathcal{K}| \geq |\mathcal{M}|$ (zakładając, że wszystkie klucze i komunikaty mają dodatnie prawdopodobieństwo).

Dowód. Ustalmy $m_0 \in \mathcal{M}$ i $k_0 \in \mathcal{K}$ o dodatnim prawdopodobieństwie. Obliczając $c_0 = s_{k_0}(m_0)$, mamy $P(C = c_0) > 0$. Jeśli szyfr jest doskonały (C niezależne od M), to

$$0 \;<\; P(C = c_0) = P(C = c_0 | M = m) \qquad \forall\, m \in \mathcal{M}\,.$$

A więc dla każdego $m \in \mathcal{M}$ istnieje klucz $k \in \mathcal{K}$ taki, że $s_k(M) = c_0$. Gdyby dla dwóch różnych m ten sam klucz dawał c_0, szyfrogramu nie można by jednoznacznie zdekodować. A więc kluczy musi być co najmniej tyle, ile możliwych komunikatów.

Szyfr doskonały jest niepraktyczny, bo do każdego szyfrowanego komunikatu trzeba niezależnie wybrać nowy klucz. Odbiorca musi go znać — problem poufnego przesłania komunikatu zamienia się na problem poufnej wymiany klucza *za każdym razem*, gdy ma dojść do szyfrowanej transmisji.

Jednoznaczność

Proces łamania szyfru na bazie statystyki (frekwencji znaków, par liter, słów itp.) opiera się na założeniu, że szyfrowany tekst jest poprawnym tekstem w jakimś języku (a więc podlega stosownej statystyce) oraz że istnieje jeden klucz, który ustala odpowiedniość między komunikatem a szyfrogramem. W takim razie dostatecznie długa próbka zaszyfrowanego tekstu powinna drogą analizy statystycznej ujawnić klucz i oryginalny komunikat.

Niech $a_1a_2...a_N \in \mathcal{A}^*$ będzie komunikatem do zaszyfrowania. Zmienna losowa $M = A_1A_2...A_N$ ma entropię $NH(\mathcal{A})$ (niezależne litery lub średnia entropia na literę razy długość dla ciągów zależnych). Szyfrogramowi odpowiada zmienna $C = C_1C_2...C_N \quad (s_K : \mathcal{A}^* \to \mathcal{B}^*)$. *Jednoznaczność* to najmniejsze N, dla którego

$$H(K|C) = 0$$

czyli liczba liter komunikatu, która jest na tyle duża, że można z niej metodami statystycznymi odtworzyć klucz K.

Zmienna C jest funkcją M i K, tak samo jak M jest funkcją (odwrotną) C i K. Stąd

$$\begin{aligned} H(K|C) &= H(K,C) - H(C) = H(M,K,C) - H(C) \\ &= H(M,K) - H(C) = H(M) + H(K) - H(C) \\ &= NH(\mathcal{A}) + \log|\mathcal{K}| - N\log|\mathcal{B}| = 0\,, \end{aligned}$$

gdzie zakładamy, że rozkłady kluczy i szyfrogramów są równomierne (użyteczne szyfry są tak właśnie skonstruowane, by utrudnić ich łamanie). Mamy więc

$$N = \frac{\log|\mathcal{K}|}{\log|\mathcal{B}| - H(\mathcal{A})}\,.$$

By zachować bezpieczeństwo, należałoby zmienić klucz po przesłaniu komunikatów o łącznej długości N.

PRZYKŁAD: Entropia języka angielskiego to ok. 1.2 bita na literę. Przyjmijmy, że stosujemy szyfr podstawieniowy, gdzie klucz jest dowolną permutacją liter i spacji. Wówczas $|\mathcal{K}| = 27!$, skąd $\log|\mathcal{K}| = 93.14$ oraz $|\mathcal{B}| = 27$, a więc jednoznaczność wynosi

$$N = \frac{\log 27!}{\log 27 - 1.2} \approx 26.2\,.$$

Szyfrowanie z publicznym kluczem

Zaszyfrować wiadomość do mnie może każdy — klucz szyfrujący jest dostępny publicznie, tak jak nr telefonu — ale odszyfrować wiadomość potrafię tylko ja. Innymi słowy: mimo znajomości klucza szyfrującego, nie można go bezpośrednio użyć do deszyfracji.

PRZYKŁAD (OSZUKIWANY!): Mając dane współczynniki wielomianu, np.

$$w(x) = 7x^6 - 4x^5 + x^3 - 2x^2 + 8x + 1\,,$$

można bez trudu policzyć np. $w(5) = 96\,991$, ale znacznie trudniej znaleźć x, dla którego $w(x) = 13\,043\,471$ $(x = -11)$. Klucz publiczny to układ współczynników w: 1, 8, −2, 1, 0, −4, 7, klucz prywatny (deszyfrujący) — ???

Każdy z uczestników (Alicja A i Bob B) ma swój klucz publiczny K_A, K_B i do niego do pary — klucz prywatny (deszyfrujący), L_A, L_B. Alicja wysyła list α do Boba, używając jego klucza publicznego:

$$\alpha \;\to\; F(K_B, \alpha) \;=\; \alpha^B \;\overset{\text{transmisja}}{\longrightarrow}\; \alpha^B \;\to\; F^{-1}(L_B, \alpha^B) \;=\; \alpha\,.$$

Podobnie Bob wysyła wiadomość do Alicji

$$\beta \;\to\; F(K_A, \beta) \;=\; \beta^A \;\overset{\text{transmisja}}{\longrightarrow}\; \beta^A \;\to\; F^{-1}(L_A, \beta^A) \;=\; \beta\,.$$

Protokół podpisu elektronicznego

— Treści dokumentu po podpisaniu nie można zmienić.
— Podpisu elektronicznego nie można skopiować i przekleić do innego pliku
— Tylko właściciel podpisu mógł go użyć.

A więc dokument i podpis muszą stanowić nierozerwalną całość. Próba korupcji uniemożliwia odczytanie dokumentu. Alicja chce wysłać do Boba podpisany przez siebie dokument. W ten sposób Bob powinien dostać gwarancję, że list pochodzi rzeczywiście od Alicji. Algorytm jest następujący:

1. Alicja przygotowuje list α i przetwarza go używając swojego klucza *prywatnego*, $\alpha \to F^{-1}(L_A, \alpha) = \alpha^A$.
2. Dokleja do tego szyfrogramu swoje nazwisko otwartym tekstem: $\alpha_1 = \alpha^A +$ `"Alicja"`.
3. Szyfruje całość publicznym kluczem Boba: $\alpha_1 \to F(K_B, \alpha_1) = \alpha_1^B$ i wysyła do Boba.
4. Bob deszyfruje otrzymają wiadomość: $\alpha_1^B \to F^{-1}(L_B, \alpha_1^B) = \alpha_1$.

5. Dowiaduje się z doklejonego podpisu kto nadał list i na tej podstawie odszukuje klucz publiczny nadawcy K_A
6. Bob wycina pierwszą (nieczytelną) część listu α^A i poddaje ją "szyfrowaniu" przy pomocy klucza K_A, $\alpha^A \to F(K_A, \alpha^A) = \alpha$.
7. Jeśli α jest czytelne, potwierdzono, że autorem jest Alicja, bo tylko ona zna klucz odwrotny do K_A.

Jak to zrobić? — algorytm RSA (Rivest–Shamir–Adelman, 1977)

- Alicja tworzy bazę dla wyznaczenia klucza publicznego: wybiera dwie liczby pierwsze p i q, np. 5 i 11 (w praktyce są to duże liczby np. 500 cyfr dziesiętnych). Oblicza $N = 5 \cdot 11 = 55$.
- Następnie oblicza $N' = (p-1)(q-1) = 4 \cdot 10 = 40$. Wybiera niedużą liczbę nieparzystą względnie pierwszą z N', np. 3. Oblicza jej odwrotność modulo N', czyli taką liczbę d, dla której $ed = 1 \pmod{N'}$. W naszym przykładzie $d = 27$, bo $3 \cdot 27 = 81 = 2 \cdot 40 + 1$.
- Klucz publiczny to para $K_A = (N, e) = (55, 3)$, klucz prywatny to $L_A = (N, d) = (55, 27)$.
- Szyfrowanie komunikatu (liczby) y, $F(K_A, y) = y^e \pmod{N}$. Długie komunikaty y dzielimy na bloki określonej długości i szyfrujemy oddzielnie. Np. szyfrując liczbę 1347, dzielimy ją na odcinki 13 i 47 i szyfrujemy jako

  $$13^3 = 52 \pmod{55}, \qquad 47^3 = 38 \pmod{55}$$

  i wysyłamy jako szyfrogram $z = 5238$.
- Alicja dekoduje szyfrogram obliczając $F^{-1}(L_A, z) = z^d \pmod{N}$. Tutaj

  $$52^{27} = 13 \pmod{55}, \qquad 38^{27} = 47 \pmod{55}.$$

Dlaczego nie można znaleźć klucza deszyfrującego (d, N) znając (e, N)?

Wystarczyłoby przecież wyznaczyć $d = e^{-1} \pmod{N'}$... Ale by znaleźć N', trzeba poznać rozkład N na czynniki p i q, a to jest trudne obliczeniowo. Dla dużych p i q liczba N może mieć około 1000 cyfr dziesiętnych, co oznacza, że proces jej faktoryzacji najlepszymi znanymi algorytmami staje się niepraktycznie długi (np. kilka tysięcy lat).

Złożoność operacji w arytmetyce

Problemy o wielomianowej (względem długości danych) złożoności czasowej:

1. Podstawowe działania: $+ \ - \ * \ / \ \mathrm{mod}$
2. NWP, NWW — algorytm Euklidesa
3. arytmetyka modularna
4. potęgowanie modularne na liczbach binarnych
5. badanie czy liczba jest pierwsza

Problemy o nieznanej złożoności czasowej (najlepsze znane algorytmy działają w czasie rosnącym wykładniczo wzgl. długości danych, choć nie wiadomo czy nie istnieją lepsze metody):

1. Faktoryzacja liczby n
2. Pierwiastkowanie kwadratowe mod N, czyli rozwiązanie $x^2 = c \pmod N$
3. Obliczanie dyskretnego logarytmu: jeśli p jest pierwsze, $\mathbb{Z}_p^*$ jest grupą cykliczną, to znaczy istnieje $a \in \{1, 2, ..., p-1\}$ takie, że $a^k \pmod p$ dla $k = 0, 1, 2...p-2$ przyjmuje wszystkie możliwe wartości. Np. dla $p = 7$ kolejne potęgi 3 modulo 7 to liczby

   $$3^0 = 1, \quad 3^1 = 3, \quad 3^2 = 2, \quad 3^3 = 6, \quad 3^4 = 4, \quad 3^5 = 5, \quad 3^6 = 1, \quad \ldots$$

   Dyskretny logarytm $\log_a m$ to najmniejsza liczba n taka, że $a^n = m \pmod p$.
4. Wyznaczanie rzędu elementu $a \in \mathbb{Z}_p^*$, to jest najmniejszego k takiego, że $a^k = 1 \pmod p$.

Bezpieczeństwo RSA bazuje na słabości znanych algorytmów faktoryzacji.

The RSA cipher relies on factoring being difficult. Up until 2007 the RSA laboratories offered rewards for factoring large challenge numbers. The RSA-B challenge involves factoring a number with B binary bits. Although the prizes have now ceased, this is still used as a test of computing power. The recent successes were

Challenge Number	Decimal digits	Factored	Prize
RSA-576	174	December, 2003	$10,000
RSA-640	193	November, 2005	$20,000
RSA-704	212	July, 2012	($30,000)
RSA-768	232	December, 2009	($50,000)
RSA-896	270		($75,000)
RSA-1024	309		($100,000)

Arytmetyczne podstawy algorytmu RSA

Podstawowe fakty o działaniach w pierścieniu reszt modulo N

LEMAT: $\mathrm{NWP}(a,b) = k$ wtedy i tylko wtedy, gdy k jest najmniejszą dodatnią liczbą całkowitą, dla której istnieją całkowite x i y takie, że $k = ax + by$.

Liczby x i y można szybko wyznaczyć przy pomocy rozszerzonego algorytmu Euklidesa, patrz np.
`http://www.algorytm.edu.pl/rozszerzony-algorytm-euklidesa.html`

WNIOSEK: Liczba a posiada odwrotność w sensie mnożenia mod N wtedy i tylko wtedy, gdy $\mathrm{NWP}(a,N) = 1$.

PRZYKŁAD: Tabliczki mnożenia mod 12 i mod 13

$\times_{12}$	1	2	3	4	5	6	7	8	9	10	11
1	1	2	3	4	5	6	7	8	9	10	11
2	2	4	6	8	10	0	2	4	6	8	10
3	3	6	9	0	3	6	9	0	3	6	9
4	4	8	0	4	8	0	4	8	0	4	8
5	5	10	3	8	1	6	11	4	9	2	7
6	6	0	6	0	6	0	6	0	6	0	6
7	7	2	9	4	11	6	1	8	3	10	5
8	8	4	0	8	4	0	8	4	0	8	4
9	9	6	3	0	9	6	3	0	9	6	3
10	10	8	6	4	2	0	10	8	6	4	2
11	11	10	9	8	7	6	5	4	3	2	1

$\times_{13}$	1	2	3	4	5	6	7	8	9	10	11	12
1	1	2	3	4	5	6	7	8	9	10	11	12
2	2	4	6	8	10	12	1	3	5	7	9	11
3	3	6	9	12	2	5	8	11	1	4	7	10
4	4	8	12	3	7	11	2	6	10	1	5	9
5	5	10	2	7	12	4	9	1	6	11	3	8
6	6	12	5	11	4	10	3	9	2	8	1	7
7	7	1	8	2	9	3	10	4	11	5	12	6
8	8	3	11	6	1	9	4	12	7	2	10	5
9	9	5	1	10	6	2	11	7	3	12	8	4
10	10	7	4	1	11	8	5	2	12	9	6	3
11	11	9	7	5	3	1	12	10	8	6	4	2
12	12	11	10	9	8	7	6	5	4	3	2	1

PRZYKŁAD: Rozwiązać kongruencję $22x = 17 \pmod{35}$.

Ponieważ $\mathrm{NWP}(22,35) = 1$, przy pomocy rozsz. alg. Euklidesa znajdujemy

$$1 \;=\; 8\cdot 22 \;-\; 5\cdot 35 \qquad \Rightarrow \qquad 8\cdot 22 \;=\; 1 \;+\; 5\cdot 35 \;=\; 1 \pmod{35}.$$

Mnożąc obydwie strony równości przez 8 otrzymujemy

$$8\cdot 22\,x \;=\; 8\cdot 17 \pmod{35} \qquad \Rightarrow \qquad x \;=\; 136 \;=\; 31 \pmod{35}$$

Rozwiązanie w postaci ogólnej: $x = 31 + 35k, \quad k = 0, 1, 2, \ldots$

Funkcja Eulera: $\varphi(N) = \left| \left\{ k: \quad 1 \le k < N, \quad \mathrm{NWP}(k,N) = 1 \right\} \right|$

$\varphi(11) = 10, \;\; \varphi(15) = 8, \;\; \varphi(12) = 4.$

Własności funkcji Eulera

(i) $\varphi(p) = p - 1$, gdy p jest liczbą pierwszą

(ii) jeśli NWP$(m, n) = 1$, to $\varphi(mn) = \varphi(m)\varphi(n)$

(ii) jeśli p jest liczbą pierwszą, $\varphi(p^n) = p^{n-1}(p-1)$

Wnioski: gdy $N = p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$

(a) $\varphi(N) = \prod_{i=1}^{k} \varphi(p_i^{n_i}) = \prod_{i=1}^{m} p_i^{n_i - 1}(p_i - 1)$

(b) $\varphi(N) = N\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right) \cdots \left(1 - \frac{1}{p_k}\right)$

Np. $12 = 2^2 \cdot 3$, więc $\varphi(12) = 12(1 - \frac{1}{2})(1 - \frac{1}{3}) = 12 \cdot \frac{1}{2} \cdot \frac{2}{3} = 4$

Twierdzenie Eulera
Jeśli liczby a i N są względnie pierwsze, wtedy $a^{\varphi(N)} = 1 \pmod{N}$.

Tzw. Małe Twierdzenie Fermata jest wnioskiem z powyższego: jeśli p jest liczbą pierwszą, dla dowolnego a mamy $a^p = a \pmod{p}$.

Przykład: Potęgowanie mod 12 i 13, $\varphi(12) = 4$, $\varphi(13) = 12$.

1	2	3	4	5	6	7	8	9	10	11
2	4	8	4	8	4	8	4	8	4	8
3	9	3	9	3	9	3	9	3	9	3
4	4	4	4	4	4	4	4	4	4	4
5	1	5	1	5	1	5	1	5	1	5
6	0	0	0	0	0	0	0	0	0	0
7	1	7	1	7	1	7	1	7	1	7
8	4	8	4	8	4	8	4	8	4	8
9	9	9	9	9	9	9	9	9	9	9
10	4	4	4	4	4	4	4	4	4	4
11	1	11	1	11	1	11	1	11	1	11

1	2	3	4	5	6	7	8	9	10	11	12
2	4	8	3	6	12	11	9	5	10	7	1
3	9	1	3	9	1	3	9	1	3	9	1
4	3	12	9	10	1	4	3	12	9	10	1
5	12	8	1	5	12	8	1	5	12	8	1
6	10	8	9	2	12	7	3	5	4	11	1
7	10	5	9	11	12	6	3	8	4	2	1
8	12	5	1	8	12	5	1	8	12	5	1
9	3	1	9	3	1	9	3	1	9	3	1
10	9	12	3	4	1	10	9	12	3	4	1
11	4	5	3	7	12	2	9	8	10	6	1
12	1	12	1	12	1	12	1	12	1	12	1

Zauważmy: jeśli N jest pierwsze, istnieją *generatory* grupy cyklicznej $\mathbb{Z}_N$. Dla $N = 13$ są to liczby 2, 6, 7 i 11.

Rząd elementu a w $\mathbb{Z}_N$, $r(a)$, to najniższa potęga k, taka że $a^k = 1 \pmod{N}$.

a	2	3	4	5	6	7	8	9	10	11	12
$r(a)$	12	3	6	4	12	12	4	3	6	12	2

$r(a)$ jest podzielnikiem $N - 1$. Generatory mają rząd równy $N - 1$.

Układy kongruencji

Istnienie rozwiązań opisuje tzw. Chińskie Twierdzenie o Resztach:

Jeśli liczby $m_1, m_2, ..., m_k$ *są parami względnie pierwsze, wtedy układ*

$$\begin{aligned} x &= a_1 \pmod{m_1} \\ x &= a_2 \pmod{m_2} \\ &\dots \\ x &= a_k \pmod{m_k} \end{aligned}$$

ma dokładnie jedno rozwiązanie w przedziale $1 \le x \le m_1 m_2 \cdots m_k$.

Istotny dla nas fakt: wszystkie rozwiązania $f(x) = 0 \pmod{N}$ spełniają układ

$$f(x) = 0 \pmod{p_1^{n_1}} \qquad \dots \qquad f(x) = 0 \pmod{p_k^{n_k}},$$

gdzie $N = p_1^{n_1} \cdots p_k^{n_k}$.

Przykład: Jakie są dwie ostatnie cyfry liczby 2^{801}?

Chcemy wyznaczyć $x = 2^{801} \pmod{100}$. Łatwo obliczyć $\varphi(100) = 40$, ale nie można stosować tw. Eulera $a^{\varphi(N)} = 1 \pmod{N}$, bo $\mathrm{NWP}(2, 100) = 2 \neq 1$. Ponieważ $100 = 2^2 \cdot 5^2$, możemy zastępczo rozważyć układ

$$\begin{aligned} x &= 2^{801} \pmod{4} \\ x &= 2^{801} \pmod{25} \end{aligned}$$

Pierwsza równość redukuje się do $x = 0 \pmod{4}$, a do drugiej można zastosować tw. Eulera, bo $\mathrm{NWP}(2, 25) = 1$:

$2^{\varphi(25)} = 2^{20} = 1 \pmod{25}$ czyli $x = 2^{801} = 2 \cdot (2^{20})^{40} = 2 \cdot 1^{40} = 2 \pmod{25}$

Z pierwszej równości mamy $x = 4k$, a więc druga sprowadza się do

$$4k = 2 \pmod{25}.$$

Ponieważ odwrotnością 4 mod 25 jest liczba 19 ($4 \cdot 19 = 76$), stąd $k = 2 \cdot 19 = 38 = 13 \pmod{25}$ i ostatecznie $x = 4 \cdot 13 = 52$.

Uzasadnienie algorytmu RSA

Alicja wybiera liczby pierwsze p i q, tworząc $N = pq$. Następnie wybiera e względnie pierwsze z $(p-1)(q-1) = \varphi(N)$ i znajduje jego odwrotność d:

$$ed = 1 \pmod{\varphi(N)}\,, \qquad \text{czyli} \quad ed = 1 + k\varphi(N)\,.$$

Korzystając z publicznego klucza Alicji (e, N), Bob koduje wiadomość y obliczając szyfrogram $y^e \pmod N$.

a) Jeśli $\mathrm{NWP}(y, N) = 1$, wtedy $y^{\varphi(N)} = 1$, a więc

$$(y^e)^d \;=\; y^{1+k\varphi(N)} \;=\; y \cdot (y^{\varphi(N)})^k \;=\; y \pmod N\,.$$

b) Gdy $\mathrm{NWP}(y, N) \neq 1$, to ponieważ $N = pq$, wtedy y musi być podzielne albo przez p albo przez q. Nie może dzielić się jednocześnie przez p i q, bo z założenia $y < N$ (ew. podział długiego komunikatu na krótsze bloki). Powiedzmy, że $y = mp$ oraz $\mathrm{NWP}(y, q) = 1$. Kongruencję $x = (y^e)^d \pmod N$ zamieniamy na układ

$$\begin{aligned} x &= y^{ed} \pmod p \\ x &= y^{ed} \pmod q \end{aligned}$$

Pierwsza równość oznacza, że $x = 0 \pmod p$, a druga daje

$$x = y^{1+k\varphi(N)} = y \cdot y^{k(p-1)(q-1)} = y \cdot y^{k(p-1)\varphi(q)} = y \cdot 1 = y \pmod q\,.$$

Równość ta zachodzi w szczególności dla $y = mp$.

W każdym z rozważanych przypadków Alicja skutecznie rozkodowuje przesłany jej szyfrogram, $(y^e)^d = y \pmod N$.

Twierdzenie Załóżmy, że posiadamy algorytm, który na podstawie publicznego klucza (e, N) potrafi wyznaczyć klucz prywatny (d, N). Wówczas przy jego pomocy można (wielomianowo w czasie) wyznaczyć faktoryzację liczby N.

Szyfr Rabina

Wybieramy dwie duże liczby pierwsze postaci $p = 4k - 1$, $q = 4m - 1$. Klucz publiczny to $N = pq$. Szyfrowanie polega na podnoszeniu komunikatu y do kwadratu mod N:

$$z \;=\; y^2 \pmod N .$$

Deszyfrowanie opiera się na poniższym lemacie i wykorzystaniu Chińskiego Twierdzenia o Resztach.

Lemat: Niech $p = 4k-1$ będzie liczbą pierwszą. Jeśli $a^2 = b \pmod p$, wówczas $a = \pm b^k \pmod p$.

Przykład: Weźmy $p = 31$, a więc $k = 8$. Obliczamy $20^2 = 28 \pmod{31}$ i zgodnie z lematem $28^8 = 20 \pmod{31}$. Można jednak sprawdzić, że także $11^2 = 28 \pmod{31}$. Jest to drugie rozwiązanie, bo $11 = -20 \pmod{31}$.

Jeśli jednak równanie $x^2 = c \pmod p$ nie ma rozwiązań, wartości $\pm c^k \pmod p$ dają się oczywiście obliczyć, ale nie spełniają one wyjściowego równania. W kontekście przesyłania komunikatu próba deszyfrowania $\pm z^k \pmod p$ produkuje nieczytelny tekst. Np. $x^2 \pmod{31}$ przyjmuje jedynie wartości 1, 2, 4, 5, 7, 8, 9, 10, 14, 16, 18, 19, 20, 25 i 28. A więc próba poszukiwania x, dla którego $x^2 = 3 \pmod{31}$ prowadzi do liczby $3^8 = 6561 = 20 \pmod{31}$, choć $20^2 = 28 \pmod{31}$, a więc wcale nie 3.

Niech teraz $N = pq = (4k-1)(4m-1)$. Bob przesyła Alicji zakodowaną wiadomość $z = y^2 \pmod N$. Alicja usiłuje wyznaczyć y szukając rozwiązań kongruencji $x^2 = z \pmod N$. Rozkłada ją na równoważny układ

$$x^2 = z \pmod p \qquad \text{i} \qquad x^2 = z \pmod q .$$

Na podstawie ostatniego lematu otrzymuje rozwiązania

$$x = \delta z^k \pmod p \qquad \text{i} \qquad x = \varepsilon z^m \pmod q ,$$

gdzie $\delta, \varepsilon = \pm 1$. Prowadzi to, z wykorzystaniem Chińskiego Twierdzenia o Resztach, do wzoru

$$y \;=\; \delta z^k + \alpha p(\varepsilon z^m - \delta z^k) \pmod N ,$$

gdzie α wyznaczone jest z równości $\alpha p + \beta q = 1$. Są to cztery różne rozwiązania dla różnych kombinacji wartości δ i $\epsilon = \pm 1$. Należy się spodziewać, że tylko jedno z nich będzie miało czytelną postać.

Twierdzenie: Algorytm łamiący szyfr Rabina pozwala sfaktoryzować N.

Kod Schmidta–Samoa

Wybieramy liczby pierwsze p i q i obliczamy

$$N = p^2 q\,, \qquad K = \mathrm{NWW}(p-1, q-1)\,, \qquad d = N^{-1} \pmod K\,.$$

Szyfrowanie: $z = y^N \pmod N$

Deszyfrowanie: $y = z^d \pmod{pq}$.

Protokoły wykorzystujące dyskretny logarytm

Potrzebna jest liczba pierwsza p i cykliczny generator γ grupy $\mathbb{Z}_p^*$.

Protokół Diffiego–Hellmana

Alicja i Bob zamierzają uzgodnić wspólny klucz szyfrujący. Liczby p i γ są dostępne publicznie.

a) Alicja wybiera prywatne $a \in \{1, 2, \ldots p-2\}$ i oblicza $A = \gamma^a$.

b) Bob podobnie wybiera $b \in \{1, 2, \ldots p-2\}$ i oblicza $B = \gamma^b$.

c) Alicja i Bob wymieniają się (otwartym kanałem) liczbami A i B. By odczytać z nich a i b, trzeba umieć obliczać dyskretne logarytmy $\log_\gamma A$ i $\log_\gamma B$.

d) Wspólnym kluczem staje się liczba $K = A^b = B^a = \gamma^{ab} \pmod p$.

Szyfrowanie: $z = Ky \pmod p$

Deszyfrowanie: $y = K^{-1} z \pmod p$

Alicja wyznaczy $K^{-1} \pmod p$ używając liczby B otrzymanej od Boba i prywatnego a:

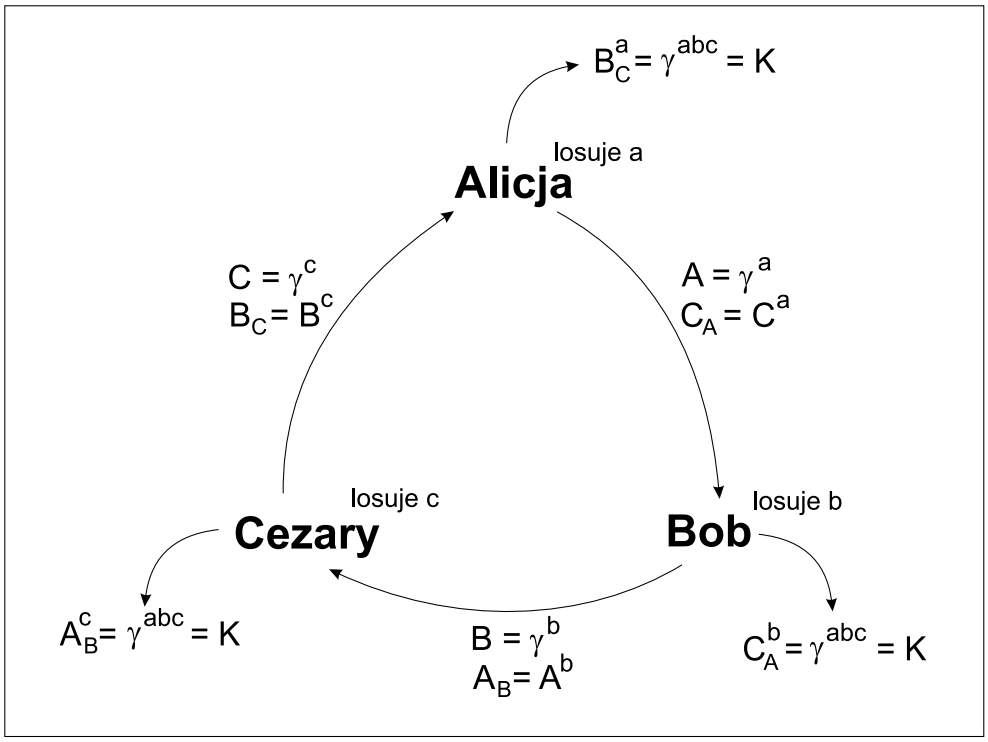
$$B^{p-1-a} = \gamma^{b(p-1-a)} = (\gamma^{p-1})^b \gamma^{-ab} = K^{-1}$$

bo $\gamma^{p-1} = \gamma^{\varphi(p)} = 1$. Bob podobnie obliczy

$$A^{p-1-b} = \gamma^{a(p-1-b)} = (\gamma^{p-1})^a \gamma^{-ab} = K^{-1}\,.$$

Przypuszczenie: Algorytm łamiący protokół D-H (odczytanie a i b z A i B) jest równoważne obliczaniu dyskretnych logarytmów.

To samo dla 3 osób (wszystkie operacje $\bmod p$):



Kod Shamira

Shamir pokazał jak wykorzystać protokół D-H do poufnej komunikacji *bez użycia publicznych kluczy* A i B. Uzgodniona jest jedynie liczba pierwsza p.

Alicja losuje liczbę a względnie pierwszą z $p-1$ i znajduje $a' = a^{-1} \pmod p$, podobnie Bob losuje b i oblicza $b' = b^{-1} \pmod p$, czyli $aa' = bb' = 1 \pmod p$. Niech y będzie komunikatem Alicji dla Boba.

1. Alicja oblicza $c = y^a \pmod p$ i wysyła do Boba.
2. Bob oblicza $d = c^b \pmod p$ i przesyła do Alicji.
3. Alicja oblicza dalej $e = d^{a'} \pmod p$ i jak poprzednio przekazuje do Boba.
4. Bob oblicza $e^{b'} = y \pmod p$.

$$e^{b'} = (d^{a'})^{b'} = (c^b)^{a'b'} = c^{a'} = (y^a)^{a'} = y \pmod p .$$

Kod ElGamala

1. Alicja tworzy i upublicznia klucz $A = \gamma^a$, wybierając $a \in \{1, 2, \ldots, p-2\}$.
2. Bob zamierza wysłać y. Wybiera $b \in \{1, 2, \ldots, p-2\}$, oblicza $B = \gamma^b$ i wysyła do Alicji parę
$$(c_0, c_1) = (B, yA^b) \pmod{p}.$$
3. Alicja oblicza $c_1 c_0^{-a} \pmod{p} = y$.

$$c_0^a = B^a = \gamma^{ba} = A^b \pmod{p}, \qquad \text{więc} \quad c_1 c_0^{-a} = yA^b A^{-b} = y \pmod{p}.$$

Łamanie kodu ElGamala jest tak samo trudne jak łamanie protokołu D-H.