Kryptografia 8

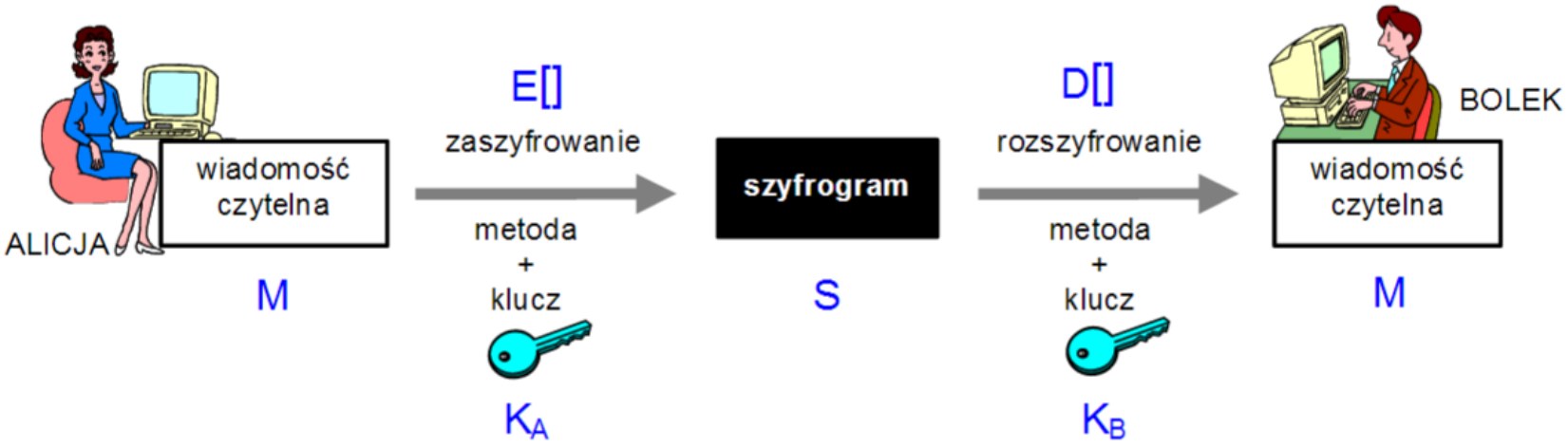
10 czerwca 2017

Kryptografia to nauka zajmująca się układaniem szyfrów.
Nazwa pochodzi z greckiego słowa: kryptos - "ukryty",
gráphein "pisać"

Szyfrowanie z kluczem

Rysunek przedstawia ogólny schemat szyfrowania z użyciem klucza, jaki stosować będziemy w niniejszym module. Użytkownicy uczestniczący w komunikacji, na tym schemacie – Alicja i Bolek, posługują się swoimi kluczami, odpowiednio – K_A oraz K_B, aby przesłać zaszyfrowaną wiadomość od Alicji do Bolka. Alicja poddaje szyfrowaniu wiadomość czytelną M z użyciem klucza szyfrowania K_A operacją $\mathbf{E_{K_A}[M]}$ uzyskując szyfrogram **S**

Następnie szyfrogram **S** jest przesyłany do Bolka, który poddaje go operacji $\mathbf{D_{K_B}[S]}$ rozszyfrowania z kluczem K_B.



Rysunek . Schemat ogólny szyfrowania z kluczem

Formalny zapis tych operacji przedstawia rysunek. Wynika z niego własność szyfrowania z kluczem: $\mathbf{D}_{K_B}[\mathbf{E}_{K_A}[\mathbf{M}]] = \mathbf{M}$.

$$E_{K_A}[M] = S \rightarrow S \rightarrow D_{K_B}[S] = M$$

Rysunek. Formalny zapis operacji szyfrowania z kluczem

W przypadku szyfrowania z kluczem spotykane są dwa schematy: szyfrowanie symetryczne i asymetryczne.

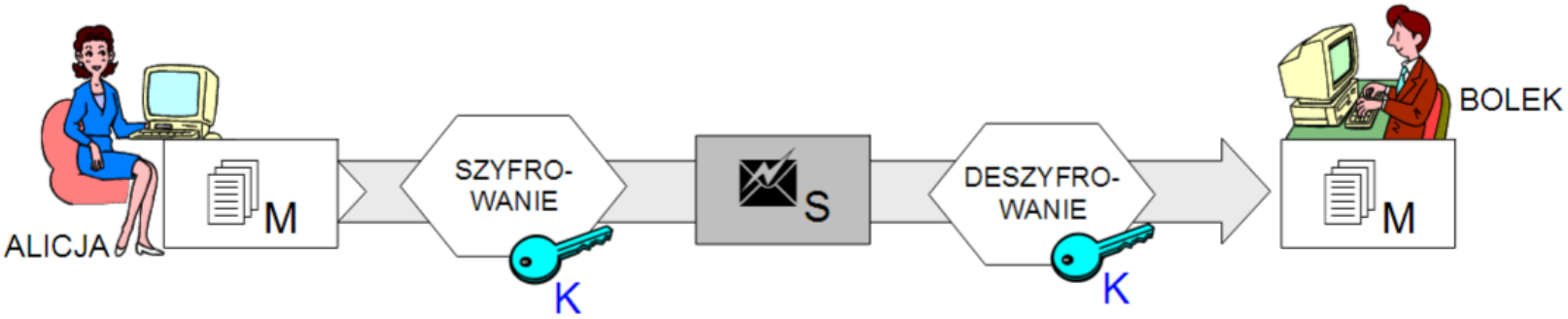
Szyfrowanie symetryczne

Szyfrowanie symetryczne jest schematem, który posiada następujące cechy

występuje wspólny dla obu uczestników komunikacji tajny klucz K_{A-B}

zapis formalny operacji szyfrowania symetrycznego ma postać:

$$E_K[M] = S \quad \rightarrow \quad S \quad \rightarrow \quad D_K[S] = M$$

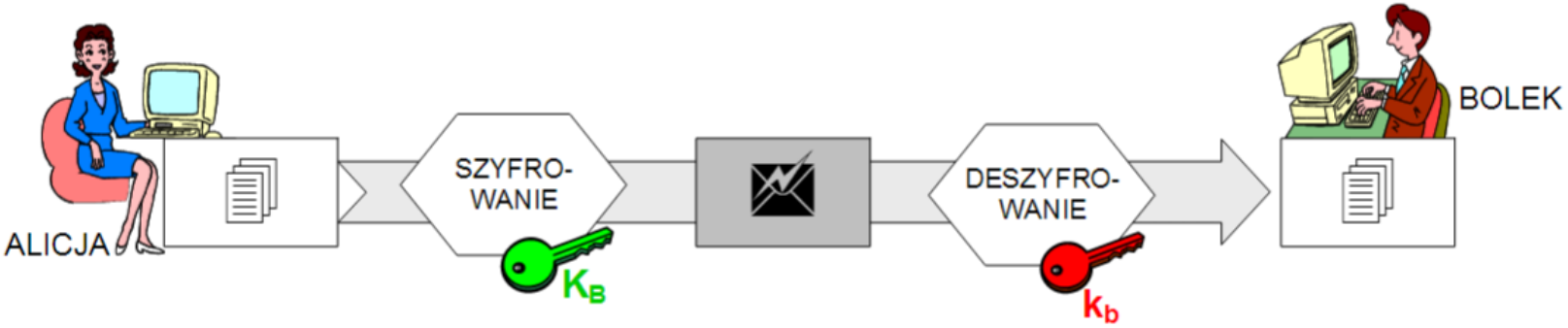


Szyfrowanie asymetryczne

Istotą szyfrowania asymetrycznego jest wyodrębnienie dwóch kluczy o odmiennych rolach: klucz prywatny i klucz publiczny. I tak przyjmiemy dalej iż odbiorca Bolek posiada parę kluczy: prywatny klucz k_b oraz publiczny klucz K_B. Z założenia klucz prywatny jest tajny znany wyłącznie właścicielowi. Publiczny klucz, natomiast, może być powszechnie znany. Aby przekazać zaszyfrowaną postać wiadomości do tego odbiorcy należy zaszyfrować wiadomość czytelną jego kluczem publicznym. Odszyfrowanie jest możliwe tylko przy użyciu klucza prywatnego, odpowiadającego użytemu uprzednio kluczowi publicznemu.

Operacje szyfrowania opisuje ogólny wzór: $E_{K_B}[M] = S$

a deszyfrację: $D_{k_b}[S] = M$



Rysunek 12. Ogólny schemat szyfrowania asymetrycznego

Istotne jest iż znajomość klucza publicznego K_B nie wystarcza do naruszenia poufności szyfrogramu uzyskanego przy zastosowaniu tego klucza.

Systemy kryptograficzne z kluczem publicznym

Funkcje jednostronne i klucze publiczne.

W 1976 roku W. Diffie i M. Hellman odkryli zupełnie nowy typ systemów kryptograficznych - systemy z publicznym kluczem.

Taki system polega na tym, że ktoś, kto zna "klucz szyfrujący" (umie zaszyfrować wiadomość) nie jest w stanie bez dodatkowej informacji znaleźć "klucza rozszyfrowującego"

(nie umie rozszyfrować innych wiadomości, zaszyfrowanych tym samym szyfrem).

System kryptograficzny składa się z:

skończonego zbioru $\mathcal{P}$ jednostek **tekstu otwartego** (np. układy po 5 liter alfabetu mającego 30 liter albo liczby od 1 do 5^{30}, albo liczby od 0 do $5^{30} - 1$);
skończonego zbioru $\mathcal{C}$ jednostek **tekstu zaszyfrowanego**;
wzajemnie jednoznacznej **funkcji szyfrującej** $f : \mathcal{P} \to \mathcal{C}$;
funkcji deszyfrującej $f^{-1} : \mathcal{C} \to \mathcal{P}$, która jest odwrotna do funkcji szyfrującej.

Nadawca tekstu zaszyfrowanego zna $\mathcal{P}$, $\mathcal{C}$ i f ; za ich pomocą dla danego tekstu otwartego $(x_1, x_2, ..., x_m) \in \mathcal{P}^m$ oblicza tekst zaszyfrowany $(f(x_1), f(x_2), ..., f(x_m)) \in \mathcal{C}^m$, który przekazuje (np. przez radio albo pocztą) odbiorcy. Odbiorca zna $\mathcal{P}$, $\mathcal{C}$ i f^{-1} ; za ich pomocą dla danego tekstu zaszyfrowanego $(f(x_1), f(x_2), ..., f(x_m)) \in \mathcal{C}^m$ oblicza tekst otwarty

$$(f^{-1}f(x_1), f^{-1}f(x_2), ..., f^{-1}f(x_m)) = (x_1, x_2, ..., x_m).$$

Funkcją progową lub **funkcją jednostronną z kluczem** nazywamy taką funkcję f, dla której obliczenie f^{-1} jest praktycznie niewykonalne jeśli zna się tylko funkcję f, ale odpowiednia osoba zna klucz - funkcję f^{-1}. Posługujący się systemem kryptograficznym z progową funkcją szyfrującą może ogłosić, jak szyfrować komunikaty, a mimo to nikt niepowołany nie będzie umiał rozszyfrować kierowanych do niego tekstów zaszyfrowanych, przynajmniej przez pewien czas.

Podobne zastosowanie mają **funkcje jednostronne**, dla których nikt nie zna funkcji odwrotnej i nie da się jej w rozsądnym czasie obliczyć. R. M. Needham zaproponował system ochrony kont komputerowych hasłami, przy którym komputer nie przechowuje danych o użytkownikach, a tylko zaszyfrowane funkcją jednostronną hasła. Hacker nie jest w stanie podszyć się pod użytkownika - nie może rozszyfrować zaszyfrowanych haseł. Rzeczywiście działający system tego typu stworzył G.B. Purdy w 1974 roku. Oryginalne hasła i ich zaszyfrowane wersje są elementami ciała $\mathbb{Z}_p$ dla dużej liczby pierwszej p. Funkcją jednostronną $f : \mathbb{Z}_p \to \mathbb{Z}_p$ był wielomian wysokiego stopnia. Purdy użył liczby $p = 2^{64} - 59$ i wielomianu $f(x) = x^{2^{24}+17} + a_1 x^{2^{24}+3} + a_2 x^3 + a_3 x^2 + a_4 x + a_5$, z dziewiętnastocyfrowymi współczynnikami a_i. Nawet znając liczbę p i wielomian f, trudno wykrzesać z siebie zapał do rozwiązywania równań $f(x) = b$.

Here is an example of a *secret key cryptosystem*. Let p be an odd prime, and let e be an integer such that $(e, p-1) = 1$. Suppose that the plaintext P is an integer such that $0 < P < p$. Let the ciphertext C be the least nonnegative residue of P^e modulo p, that is, we construct C by the rule

$$C \equiv P^e \pmod{p}$$

and

$$0 < C < p.$$

The encrypting key for this cipher consists of the prime number p and the integer e. To decrypt this cipher, we use elementary number theory. Since $(e, p-1) = 1$, there exists an integer d such that $ed \equiv 1 \pmod{p-1}$. It is easy to compute d. We can use the Euclidean algorithm, for example. The decrypting key consists of the prime p and the integer d. Since $ed = 1 + (p-1)k$ for some integer k, and since $P^{p-1} \equiv 1 \pmod{p}$ by Fermat's theorem, it follows that

$$C^d \equiv P^{ed} \equiv P^{1+(p-1)k} \equiv P(P^{p-1})^k \equiv P \pmod{p}.$$

The encrypting key for this cipher consists of the prime number p and the integer e. To decrypt this cipher, we use elementary number theory. Since $(e, p-1) = 1$, there exists an integer d such that $ed \equiv 1 \pmod{p-1}$. It is easy to compute d. We can use the Euclidean algorithm, for example. The decrypting key consists of the prime p and the integer d. Since $ed = 1 + (p-1)k$ for some integer k, and since $P^{p-1} \equiv 1 \pmod{p}$ by Fermat's theorem, it follows that

$$C^d \equiv P^{ed} \equiv P^{1+(p-1)k} \equiv P(P^{p-1})^k \equiv P \pmod{p}.$$

Thus, we can decrypt the ciphertext C by computing the least nonnegative residue of C^d modulo p. An enemy who learns the encrypting key will break the cipher.

Thus, we can decrypt the ciphertext C by computing the least nonnegative residue of C^d modulo p. An enemy who learns the encrypting key will break the cipher.

For example, if $p = 17$ and $e = 3$, then the plaintext $P = 10$ is encrypted as

$$P^3 = 10^3 \equiv 14 \pmod{17},$$

and so the ciphertext is $C = 14$. Since $3 \cdot 11 \equiv 1 \pmod{16}$, it follows that $d = 11$ is a decrypting key. We observe that

$$C^{11} = 14^{11} \equiv 10 = P \pmod{17}.$$

There is a more sophisticated idea in cryptography that produces secure ciphers even if the encrypting key is known. Indeed, the encrypting key can be made public, so that anyone can encrypt and send a message, but the decrypting key cannot be computed from knowledge of the encrypting key.

Najsłynniejszy ostatnio system kryptograficzny z publicznym kluczem (jawnymi funkcjami szyfrującymi, które są progowe), system RSA, został zaproponowany przez słynnego popularyzatora matematyki, Martina Gardnera w 1977 r. na łamach Scietific American. Rok później R.L. Rivest, A. Shamir i L.M. Adleman ogłosili dokładne zasady tworzenia takich systemów kryptograficznych.

Bezpieczeństwo algorytmu RSA zostało oparte na problemie faktoryzacji liczb całkowitych.

Klucze:

generujemy liczby pierwsze p, q, wyznaczamy

$$N = pq$$

i szukamy takich liczb

d (klucz prywatny)
e (klucz publiczny),

że

$$ed \equiv 1 \bmod (p-1)(q-1).$$

Wykonanie operacji prywatnej może być przyspieszone.
Wystarczy wykorzystać twierdzenie chińskie o resztach:

$$C_p \equiv M^{d \bmod (p-1)} \bmod p,$$

$$C_q \equiv M^{d \bmod (q-1)} \bmod q,$$

$$\begin{cases} C \equiv C_p \bmod p, \\ C \equiv C_q \bmod q. \end{cases}$$

Dobór kluczy

p, q – losowo wybrane duże liczby pierwsze

$n = p \cdot q$ – moduł

e – liczba względnie pierwsza z (p-1)(q-1)

d – liczba wyznaczona tak, że zachodzi $(e \cdot d) \; mod \; (p-1)(q-1) = 1$

$d = e^{-1} \; mod \; (p-1)(q-1)$

$k_a \Rightarrow n, d$

$K_A \Rightarrow n, e$

Szyfrowanie

$$E_{K_A}[M] = M^e \; mod \; n = S$$

Deszyfrowanie

$$D_{k_a}[S] = S^d \; mod \; n = M^{ed} \; mod \; n = M^1$$ (z twierdzenia Eulera)

Ogólny schemat pracy algorytmu RSA

liczba względnie pierwsza z (p-1) i (q-1) to inaczej Wykładnik Uniwersalny: dzieli się tylko przez 1, siebie oraz (p-1) i (q-1), czyli Najmniejsza Wspólna Wielokrotność, albo inaczej: Największy Wspólny Dzielnik (e, (p-1)(q-1)) = 1

odwrotność e można łatwo wyznaczyć rozszerzonym algorytmem Euklidesa.

This is called a *public key cryptosystem*. Here is an example. We choose two different large primes p and q, and let

$$m = pq.$$

This is called a *public key cryptosystem*. Here is an example. We choose two different large primes p and q, and let

$$m = pq.$$

Since we know p and q, it is easy to calculate $\varphi(m) = (p-1)(q-1)$. Pick an integer e that is relatively prime to $\varphi(m)$. We publish the numbers m and e. The plaintext must be a positive integer P that is less than m and relatively prime to m If m is a large number, then almost all positive integers less than m are relatively prime to m (Exercise 4), so we can assume that $(P, m) = 1$. The ciphertext will be the unique integer C such that

$$C \equiv P^e \pmod{m}$$

It is important to note that we disclose neither $\varphi(m)$ nor the prime factors p and q of m. These are kept secret. However, since *we* know $\varphi(m)$, it is easy, by using the Euclidean algorithm, for example, to compute an integer d such that

$$ed \equiv 1 \pmod{\varphi(m)},$$

that is,

$$ed = 1 + \varphi(m)k$$

for some integer k. To decrypt the ciphertext C, we simply compute the least nonnegative residue of

$$C^d \pmod{m}.$$

Since $(P, m) = 1$, Euler's theorem tells us that

$$C^d \equiv P^{ed} \equiv P^{1+\varphi(m)k} \equiv P \pmod{m}.$$

It is important to note that we disclose neither $\varphi(m)$ nor the prime factors p and q of m. These are kept secret. However, since *we* know $\varphi(m)$, it is easy, by using the Euclidean algorithm, for example, to compute an integer d such that

$$ed \equiv 1 \pmod{\varphi(m)},$$

that is,

$$ed = 1 + \varphi(m)k$$

for some integer k. To decrypt the ciphertext C, we simply compute the least nonnegative residue of

$$C^d \pmod{m}.$$

Since $(P, m) = 1$, Euler's theorem tells us that

$$C^d \equiv P^{ed} \equiv P^{1+\varphi(m)k} \equiv P \pmod{m}.$$

The decryption key requires the integers d and m. It is not enough to know e and m. To compute d, one must know both e and $\varphi(m)$. Since $\varphi(m) = (p-1)(q-1)$, this requires a knowledge of the primes p and q such that $m = pq$, that is, we must be able to factor m.

We know the prime factors p and q, and so we can compute $\varphi(m)$, but an opponent who wants to intercept and decrypt the message will fail, since he does not know the primes and cannot factor m. Indeed, the following result shows that knowing $\varphi(m)$ is equivalent to knowing the prime factors of m.

Theorem *Let m be an integer that is the product of two prime numbers. The prime divisors of m are the roots of the quadratic equation*

$$x^2 - (m + 1 - \varphi(m))x + m = 0,$$

and so $\varphi(m)$ determines the prime factors of m.

Proof. If $m = pq$, then

$$\varphi(m) = (p-1)(q-1) = pq - p - q + 1 = m - p - \frac{m}{p} + 1,$$

and so

$$p - (m + 1 - \varphi(m)) + \frac{m}{p} = 0.$$

Equivalently, p and q are the solutions of the quadratic equation

$$x^2 - (m + 1 - \varphi(m))x + m = 0.$$

This completes the proof. □

For example, if $m = 221$ and $\varphi(m) = 192$, then the quadratic equation

$$x^2 - 30x + 221 = 0$$

has solutions $x = 13$ and $x = 17$, and $221 = 13 \cdot 17$.

This method, known as the *RSA cryptosystem*, is called a *public key cryptosystem*, since the encryption key is made available to everyone, and the encrypted message can be transmitted through public channels. Only the possessor of the prime factors of m can decrypt the message. RSA is simple, but useful, and is the basis of many commercially valuable cryptosystems.

Oto system RSA:

Twórca systemu wybiera dwie duże liczby pierwsze p i q. Na mocy twierdzenia chińskiego o resztach

$$\mathbb{Z}_{pq} \cong \mathbb{Z}_p \times \mathbb{Z}_q.$$

W szczególności $\mathbb{Z}_{pq}^* \cong \mathbb{Z}_p^* \times \mathbb{Z}_q^*$ jest grupą mającą $\varphi(pq) = (p-1)(q-1)$ elementów. Z twierdzenia Eulera wynika, że każdy element tej grupy podniesiony do potęgi $(p-1)(q-1)$ daje w wyniku jedynkę tej grupy, czyli jedynkę pierścienia $\mathbb{Z}_{pq}$, czyli resztę 1 z dzielenia przez $r = pq$. Mając k użytkowników $A_1, A_2, ..., A_k$, wybieramy k liczb $s_1, s_2, ..., s_k$ względnie pierwszych z $(p-1)(q-1)$. Dla każdej z nich obliczamy element odwrotny t_i w $\mathbb{Z}_{(p-1)(q-1)}$, to znaczy obliczamy liczby $t_1, t_2, ..., t_k$ dla których $s_i t_i \equiv 1 (\mathrm{mod} (p-1)(q-1))$. Wtedy dla dowolnej liczby x względnie pierwszej z $r = pq$ zachodzi równość

$$(x^{s_i})^{t_i} \equiv x (\mathrm{mod}\, r).$$

Rzeczywiście, $s_i t_i = 1 + m_i(p-1)(q-1)$, więc $(x^{s_i})^{t_i} = x^{1+m_i(p-1)(q-1)} = x \cdot (x^{(p-1)(q-1)})^{m_i} \equiv x$. Podnoszenie do potęgi s_i modulo r będzie funkcją szyfrującą użytkownika A_i, a podnoszenie do potęgi t_i modulo r będzie funkcją deszyfrującą tegoż użytkownika. Działanie systemu wygląda tak: każdy użytkownik A_i otrzymuje liczbę r (tym samym zna zbiór jednostek tekstu otwartego i zbiór jednostek tekstu zaszyfrowanego), wszystkie klucze szyfrujące $s_1, s_2, ..., s_k$ i swój klucz deszyfrujący t_i. Chcąc wysłać tajny list do uzytkownika A_j, użytkownik A_i szyfruje go kluczem s_j i tylko użytkownik A_j może taki list rozszyfrować.

Jednostronność funkcji szyfrującej bierze się stąd, że nie ma szybkich algorytmów rozkładania na czynniki pierwsze, które pozwoliłyby na podstawie r obliczyć p, q, zatem $(p-1)(q-1)$ i t_j na podstawie s_j.

TWIERDZENIE. Istnieje wzajemnie jednoznaczna odpowiedniość między rozkładami $n = ab$ liczby nieparzystej n, a przedstawieniami $n = t^2 - s^2$ liczby n jako różnicy kwadratów.

$$\begin{cases} t = \frac{a+b}{2} \\ s = \frac{a-b}{2} \end{cases} \qquad \begin{cases} a = t + s \\ b = t - s \end{cases}.$$

Szanse znalezienia takiego rozkładu są szczególnie duże gdy czynniki a i b różnią się nieznacznie: wtedy liczba $s = \frac{a-b}{2}$ jest mała, a t nieznacznie przewyższa $\sqrt{n}$.

Przykład. Rozłożymy na czynniki liczbę 200819.
$\sqrt{200819} \approx 448.13$, zaczynamy od 449:

t	t^2	$t^2 - n$	$\sqrt{t^2 - n}$
449	201601	782	27.964
450	202500	1681	41.0

Zatem $450^2 - 41^2 = 200819$ i $200819 = (450 + 41)(450 - 41) = 491 \cdot 409$.

Wystarczy ograniczyć się do obliczania reszt z dzielenia przez n. Jeśli uda się znaleźć rozwiązanie kongruencji $t^2 \equiv s^2 (\bmod n)$, to $t^2 - s^2 = kn$ dla pewnego k. Jeśli k jest małe, a t nie przystaje do $\pm s (\bmod n)$, to $\mathrm{NWD}(t+s, n)$ i $\mathrm{NWD}(t-s, n)$ są dzielnikami liczby n.

Polynomials and Primitive Roots

Let m be a positive integer greater than 1, and a an integer relatively prime to m. The *order of a modulo m*, denoted by $\mathrm{ord}_m(a)$, is the smallest positive integer d such that $a^d \equiv 1 \pmod{m}$.

$\mathrm{ord}_m(a)$ is a divisor of the Euler phi function $\varphi(m)$.

The order of a modulo m is also called the *exponent* of a modulo m.

The integer a is called a *primitive root modulo* m if a has order $\varphi(m)$. In this case, the $\varphi(m)$ integers $1, a, a^2, \ldots, a^{\varphi(m)-1}$ are relatively prime to m and are pairwise incongruent modulo m.
For example, 3 is a primitive root modulo 7.

Similarly, 3 is a primitive root modulo 10, since $\varphi(10) = 4$ and

$$\begin{aligned} 3^0 &\equiv 1 \pmod{10}, \\ 3^1 &\equiv 3 \pmod{10}, \\ 3^2 &\equiv 9 \pmod{10}, \\ 3^3 &\equiv 7 \pmod{10}, \\ 3^4 &\equiv 1 \pmod{10}. \end{aligned}$$

Some moduli do not have primitive roots.

ElGamal encryption

asymmetric key encryption algorithm
public-key cryptography

The algorithm

ElGamal encryption consists of three components:

the key generator,
the encryption algorithm,
and the decryption algorithm.

Key generation The key generator works as follows:

Alice generates an efficient description of a cyclic group G of order q with generator g.

Alice chooses an x randomly from $\{1, \ldots, q-1\}$.

Alice computes $h := g^x$.

Alice publishes h, along with the description of G, q, g, as her **public key**.

Alice retains x as her **private key**, which must be kept secret.

Encryption

The encryption algorithm works as follows: to encrypt a message m to Alice under her public key (G, q, g, h),

- Bob chooses a random y from $\{1, \ldots, q-1\}$, then calculates $c_1 := g^y$.
- Bob calculates the shared secret $s := h^y := g^{xy}$.
- Bob maps his secret message m onto an element m' of G.
- Bob calculates $c_2 := m' \cdot s$.
- Bob sends the ciphertext $(c_1, c_2) = (g^y, m' \cdot h^y) = (g^y, m' \cdot g^{xy})$ to Alice

Decryption

The decryption algorithm works as follows: to decrypt a ciphertext (c_1, c_2) with her private key x,

Alice calculates the shared secret $s := {c_1}^x$

and then computes $m' := c_2 \cdot s^{-1}$ which she then converts back into the plaintext message m,

where s^{-1} is the inverse of s in the group G.

The decryption algorithm produces the intended message, since

$$c_2 \cdot s^{-1} = m' \cdot h^y \cdot (g^{xy})^{-1} = m' \cdot g^{xy} \cdot g^{-xy} = m'.$$

Algorytm ElGamala (ELG)

Algorytm ten został opublikowany w 1985 roku, ale nie jest chroniony patentem. Brak również w jego przypadku ograniczeń eksportowych USA – wykorzystuje koncepcję (i patent) Diffiego-Helmana lecz ów patent wygasł w 1997 r. Szyfrowanie wymaga każdorazowo losowo wybranej wartość *k*, dlatego też ten sam tekst jawny każdorazowo daje inny szyfrogram. Niestety wadą tego algorytmu jest fakt, iż szyfrogram jest dwukrotnie dłuższy od tekstu jawnego.

Generowanie kluczy przebiega następująco:

wybieramy losowo liczbę pierwszą p

wykorzystujemy multiplikatywną grupę modulo p – $\mathbb{Z}_p^*$

gdzie p jest liczbą pierwszą, a $\mathbb{Z}_p$ jego ciałem skończonym ($\mathbb{Z}/p\mathbb{Z}$)

wybieramy liczbę g, która jest elementem pierwotnym (generatorem) grupy $\mathbb{Z}_p^*$

generator – generuje ciąg 1, g, g^2, g^3, ...

z którego tylko skończenie wiele należy do $\mathbb{Z}_p^*$ (potem zaczną się powtarzać modulo p)

w ogólności mamy q elementów: 1, g, g^2, ... , g^{q-1} (g^q mod $p = 1$)

istnieje przynajmniej jedno g generujące całą grupę! (tzn. $q = p$-1)

czyli zamiast 1, ..., p-1 możemy grupę traktować jako 1, g, g^2, ... , g^{p-2}

W dalszej kolejności:

- wybieramy losowo liczbę $x < p$
- obliczamy $y = g^x \bmod p$
- klucz publiczny stanowią y, g i p – zarówno g, jak i p mogą być wspólnie wykorzystywane przez grupę użytkowników ($\bmod p$)
- kluczem prywatnym jest x

Szyfrowanie przebiega następująco:

wybieramy losowo liczbę k względnie pierwszą z p

obliczamy $a = g^k \bmod p$

obliczamy $b = y^k \cdot \mathrm{M} \bmod p$

szyfrogram to para (a, b)

Deszyfrowanie przebiega następująco:

$\mathrm{M} = b/a^x \bmod p$

ponieważ $a^x \equiv g^{kx} \bmod p$

$b/a^x \equiv y^k \cdot \mathrm{M}/a^x \equiv g^{xk} \cdot \mathrm{M}/g^{kx} \equiv \mathrm{M} \bmod p$