Matematyka Podpisu Elektronicznego 20.10.04

Wykład 3. Klasyczne kryptosystemy asymetryczne.

Stefan Dziembowski stefan-mpe@dziembowski.net

Streszczenie. Definiujemy pojęcie reszty kwadratowej i symbolu Jacobiego. Pokazujemy, że operacja szyfrowania RSA zachowuje wartość symbolu Jacobiego. Wprowadzamy kryptosystem Rabina

Szyfr (i schemat szyfrowania) RSA w takiej formie jak zaprezentowana na ostatnim wykładzie nie nadaje się do natychmiastowego użycia, tak jak każdy szyfr w którym szyfrowanie jest deterministyczne (z powodów podanych w Rozdziale 3.2 Wykładu 2). Na przyszłych wykładach zajmiemy się przekształceniem RSA do takiej formy, by było ono używalne w praktyce. Najpierw jednak przedstawimy dalszy ciąg klasycznego „teorio-liczbowego" wykładu kryptografii.

6 Grupy cykliczne

Przypomnijmy, że skończona grupa G jest cykliczna wtedy i tylko wtedy gdy istnieje g takie, że $G = \left\{g^0, \ldots, g^{|G|-1}\right\}$. Mam miejsce następujący fakt.

Twierdzenie 1 *Dla dowolnej liczby pierwszej p grupa Z_p^* jest cykliczna.*

(Dowód pomijamy).

7 Reszty kwadratowe

Definicja 2 *Liczba x jest* resztą kwadratową modulo n *jeśli istnieje $y \in Z_n^*$ takie, że $x = y^2 \bmod n$. Zbiór wszystkich takich x oznaczamy* QR_n. *Ponadto* $\overline{\mathrm{QR}}_n := Z_n^* \setminus \mathrm{QR}_n$.

Nieformalnie mówiąc: reszty kwadratowe to takie liczby, które mają pierwiastek kwadratowy modulo n. Na przykład, resztami kwadratowymi modulo 9 są: $1, 4$ i 7. Pokażemy teraz, że jeśli p jest nieparzystą liczbą pierwszą (to znaczy: $p > 2$), to równanie $x^2 = y$ ma albo 0 albo 2 rozwiązania w Z_p^*. Inaczej mówiąc funkcja dana wzorem $f(x) = x^2 \bmod p$ skleja dokładnie dwa elementy Z_p^*, to znaczy dla dowolnego x istnieje dokładnie jeden $x'(\neq x)$ taki, że

$$f(x) = f(x').$$

Konkretnie: jest to $p - x$ (ponieważ p jest nieparzysta, to $x \neq p - x$). Rzeczywiście: (1) jak łatwo sprawdzić $x^2 = (p-x)^2 \pmod p$ a ponadto (2) jeśli $x^2 = (x')^2 \pmod p$, to $(x - x')(x + x')$ jest podzielne przez p, a ponieważ p jest liczbą pierwszą, to albo $x = x'$ albo $x = -x' \pmod p$. Zatem zbiór reszt kwadratowych ma moc dokładnie $\left|\vec{f}(Z_p)\right| = (p-1)/2$. (Załóżmy teraz, że wszystkie operacje wykonywane są w Z_p.) Jeśli g jest generatorem to oczywiście parzyste potęgi generatora $(g^0, g^2, g^4, \ldots, g^{p-3})$ są resztami kwadratowymi. Ponieważ jest ich dokładnie $(p-1)/2$ (i są one parami różne, bo ich potęgi są mniejsze od rzędu grupy) to są to *wszystkie* takie reszty. Zatem mamy:

$$\mathrm{QR}_p = \left\{g^{2i} : i = 0, \ldots, \frac{p-3}{2}\right\}.$$

Pokażemy teraz wydajne kryterium sprawdzania, czy liczba x jest resztą kwadratową modulo p.[1] Zauważmy, że jeśli element $x \in \mathrm{QR}_p$ podniesiemy do potęgi $(p-1)/2$, to uzyskamy 1 (bo $x^{(p-1)/2} = y^{p-1} = 1$, gdzie y jest pewnym elementem Z_p^*, a ostatnia równość wynika z Małego Twierdzenia Fermata, patrz Rozdział 5.1, Wykład 2). Jeśli natomiast $x \notin \mathrm{QR}_p$, to $x = g^{2i+1} = g \cdot g^{2i}$ (gdzie i jest pewną liczbą naturalną). Zatem $x^{(p-1)/2} = g^{(p-1)/2} \cdot g^{p-1} = g^{(p-1)/2} \neq 1$. Można zresztą łatwo pokazać, że $g^{(p-1)/2} = -1$. Aby to zobaczyć zauważmy, że $f(g^{(p-1)/2}) = 1$. Przypomnijmy, że f skleja dokładnie dwa elementy. W tym przypadku oczywiście $f^{-1}(1) = \{1, -1\}$. Ponieważ $g^{(p-1)/2} \neq 1$ (bo g jest generatorem), to $g^{(p-1)/2}$ musi być równe -1. Mamy zatem następujący fakt.

Lemat 3 *Dla dowolnej liczby pierwszej p istnieje wydajny algorytm sprawdzania czy dana liczba x jest resztą kwadratową modulo p.*

Dowód. Wystarczy obliczyć $g^{(p-1)/2} \bmod p$ i sprawdzić, czy wyszło $+1$, czy -1. □

Definicja 4 *Symbol Legendre'a $\left(\frac{a}{n}\right)$ (gdzie n jest nieparzystą liczbą pierwszą) jest funkcją $\mathbf{Z} \to \{-1, 0, +1\}$ zdefiniowaną w następujący sposób:*

$$\left(\frac{a}{p}\right) := \begin{cases} 0 & \text{jeśli } a \bmod n = 0 \\ +1 & \text{jeśli } a \bmod n \in \mathrm{QR}_n \\ -1 & \text{jeśli } a \bmod n \notin \mathrm{QR}_n. \end{cases}$$

Inaczej mówiąc: $\left(\frac{a}{n}\right) = a^{(n-1)/2} \bmod n$ (jeśli tylko $n > 2$). Symbol Legendre'a uogólnia się (na przypadki kiedy n jest dowolną nieparzystą liczbą naturalną) w następujący sposób.

Definicja 5 *Niech $n = p_1^{\alpha_1} \cdot \cdots \cdot p_k^{\alpha_k}$, gdzie $p_1, \ldots, p_k$ są (parami różnymi) nieparzystymi liczbami pierwszymi a $\alpha_1, \ldots, \alpha_k$ są liczbami naturalnymi. Symbol Legendre'a $\left(\frac{a}{n}\right)$ jest funkcją $\mathbf{N} \to \{-1, 0, +1\}$ zdefiniowaną jak następuje:*

$$\left(\frac{a}{n}\right) := \left(\frac{a}{p_1}\right)^{\alpha_1} \cdot \cdots \cdot \left(\frac{a}{p_k}\right)^{\alpha_k}.$$

Zwróćmy uwagę, że symbol Jacobiego nie daje informacji na temat tego, czy liczba jest resztą kwadratową (np. $\left(\frac{2}{9}\right) = \left(\frac{2}{3}\right)^2 = 1$, ale 2 nie jest resztą kwadratową modulo 9). Mamy następujący fakt.

Twierdzenie 6 *Istnieje wydajny algorytm obliczający (dla danego n i a) wartość $\left(\frac{a}{n}\right)$ (nawet jeśli nie jest znana faktoryzacja n).*

Dowód pomijamy (znajduje się np. w [Sti02]).

8 Bezpieczeństwo RSA

Ma miejsce następujący fakt.

Lemat 7 *Dla dowolnych a, n i i mamy:*

$$\left(\frac{a^i}{n}\right) = \left(\frac{a}{n}\right)^i.$$

[1] Pomysł polegający na obliczeniu j takiego, że $g^j = x$ i sprawdzeniu czy j jest parzysta, odpada, bo nie jest jasne jak takie j obliczyć (zresztą jak się niebawem okaże wygląda na to, że się nie da).

Dowód. Łatwy wniosek z tego, że dla dowolnych a i b mamy

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right)\left(\frac{b}{n}\right)$$

(a to z kolei wynika z definicji symbolu Jacobiego). □

Przypomnijmy, że w szyfrowanie RSA polega na podniesieniu do potęgi e, która jest nieparzysta, bo musi być względnie pierwsza z $\varphi(n) = (p-1)(q-1)$. Zatem mamy:

$$\left(\frac{\mathcal{E}^{\mathrm{RSA}}_{n,e}(x)}{n}\right) = \left(\frac{x^e}{n}\right) = \left(\frac{x}{n}\right)^e = \left(\frac{x}{n}\right)$$

(bo podnoszenie do nieparzystej potęgi jest identycznością na zbiorze $\{-1, 0, 1\}$). Stąd:

Wniosek 8 ([Lip81]) *Szyfrowanie RSA zachowuje wartość symbolu Jacobiego.*

Ponieważ wartość symbolu Jacobiego jest wydajnie obliczalna (Twierdzenie 6) można więc (na podstawie szyfrogramu $\mathcal{E}^{\mathrm{RSA}}_{n,e}(x)$ i modułu n) obliczyć wartość symbolu Jacobiego wiadomości x. Nie wydaje się jednak, by stanowiło to problem w większości praktycznych zastosowań.

9 Kryptosystem Rabina

9.1 Wstęp

Jak się za chwilę okaże, znajdowanie pierwiastków modulo n (gdzie $n = pq$ jest modułem RSA) jest łatwe jeśli znamy faktoryzację n i trudne (tak samo trudne jak sama faktoryzacja) gdy jej nie znamy. Stąd pomysł Rabina [Rab79] na kryptosystem w którym szyfrowanie polega na podnoszeniu do kwadratu modulo n. Zauważmy, że równanie

$$x^2 = y \pmod{n} \tag{1}$$

ma albo 0 albo 4 rozwiązania w Z_n^*. Jest tak dlatego, że równanie (1) jest (na mocy Chińskiego Twierdzenia o Resztach) równoważne układowi równań

$$\begin{cases} x^2 = y \pmod{p} \\ x^2 = y \pmod{q}, \end{cases} \tag{2}$$

a zatem

- spełniają je tylko takie x, które są jednocześnie resztami kwadratowymi modulo p i modulo q (jest ich dokładnie 1/4),
- jeśli x taki, że $x = x_p \pmod{p}$ i $x = x_q \pmod{q}$ spełnia to równanie, to spełniają je wszystkie x takie, że $x = \pm x_p \pmod{p}$ i $x = \pm x_q \pmod{q}$ (jest ich dokładnie 4).

Zauważmy, że mając jedno rozwiązanie x równania (1) możemy łatwo uzyskać drugie (nawet bez znajomości faktoryzacji n). Jest to mianowicie $n - x$. Natomiast obliczenie pozostałych dwóch pierwiastków jest co najmniej tak trudne[2] jak faktoryzacja n. Jak łatwo bowiem zauważyć jeśli mamy $x' \neq \pm x \pmod{p}$, to albo $x = x' \pmod{p}$ albo $x = x' \pmod{q}$ (bo inaczej $x = -x' \pmod{n}$). Zatem wystarczy obliczyć $\gcd(x, x')$ i otrzymamy albo p albo q. Stąd już niedaleko do wykazania, że pierwiastkowanie w Z_n^* jest co najmniej tak samo trudne jak faktoryzacja.

[2] Za chwilę (Lemat 9) pokażemy, że w rzeczywistości jest obliczeniowo równoważne.

Lemat 9 *Niech $\mathcal{A}$ będzie WAPem który na wejściu $n = pq$ i $y \in \mathrm{QR}_n$ zwraca x taki, że $x^2 = y \pmod n$. Wówczas istnieje WAP $\mathcal{B}$ który na wejściu n oblicza p oraz q.*

Dowód. Algorytm $\mathcal{B}$ na wejściu n wybiera losowy element $x \in Z_n^*$ i uruchamia algorytm $\mathcal{A}$ na argumentach n i $x^2 \pmod n$. Algorytm $\mathcal{A}$ zwraca pierwiastek x' liczby x^2 (modulo n). Z poprzednich uwag wynika, że jeśli

$$x \neq \pm x' \pmod n, \tag{3}$$

to algorytm $\mathcal{B}$ bez trudu obliczy faktoryzację n. W przeciwnym przypadku $\mathcal{B}$ rozpoczyna procedurę od początku.

Jeśli algorytm $\mathcal{A}$ zwrócił x', to szansa, że zaszło (3) wynosi dokładnie $1/2$, więc oczekiwana liczba powtórzeń wynosi 2. □

Pokażemy teraz fakt odwrotny, to znaczy, że umiejętność faktoryzacji n pozwala obliczać pierwiastki modulo n. Zaprezentujemy mianowicie wydajny algorytm znajdujący pierwiastek kwadratowy $x \in \mathrm{QR}_n$ modulo n przy znajomości p i q (co umożliwi nam odszyfrowywanie). Z Chińskiego Twierdzenia o resztach wystarczy oczywiście skonstruować algorytm znajdujący pierwiastki modulo p.

Lemat 10 *Istnieje algorytm (działający w oczekiwanym czasie wielomianowym) pobierający na wejściu p i $x \in \mathrm{QR}_p$ i zwracający y takie, że $x = y^2 \pmod p$.*

Dowód. Rozpatrujemy 2 przypadki.

$p = 3 \pmod 4$ Niech $p = 4i+3$. Ponieważ (patrz Rozdział 7) mamy $x^{p-1/2} = 1 \pmod p$, więc $x^{2i+1} = 1 \pmod p$ a zatem $x^{2i+2} = x \pmod p$. Liczba $y := x^{i+1} \pmod p$ jest więc szukanym pierwiastkiem.

$p = 1 \pmod 4$ Niech $p = 4i + 1$. Stosując podobne rozumowanie jak poprzednio uzyskujemy $x^{2i} = 1 \pmod p$. I tu klops, bo $2i$ jest parzyste a naszym celem jest uzyskanie równania $x^{2j+1} = 1 \pmod p$. Stąd następujący pomysł: ponieważ $x^i \bmod n \in \{-1, 1\}$, to możemy stworzyć następującą procedurę:

1. $k := 2i$; $w := x$
2. Póki k jest parzyste powtarzaj:
   (a) $k := k/2$
   (b) jeśli
   $$w^k = -1 \pmod p, \tag{4}$$
   to zatrzymaj się z komunikatem error
3. Jeśli algorytm dotarł do tego miejsca to znaczy, że k jest nieparzyste, a zatem mamy równanie $w^{2j+1} = 1 \pmod p$ (gdzie $k = 2j + 1$) i jesteśmy w domu (bo $w^{2j+2} = 1 \pmod p$ i szukanym pierwiastkiem jest $y := w^{j+1}$).

Jedynym problemem jest to, że z bardzo dużym prawdopodobieństwem algorytm nigdy nie dotrze do Linii 3 (bo wcześniej zwróci error). Remedium na ten problem jest następujące. Wybieramy jakieś $z \in \overline{\mathrm{QR}}_p$ (w jaki sposób — o tym za chwilę). Oczywiście

$$z^{2i} = -1 \pmod p. \tag{5}$$

Za każdym razem kiedy otrzymujemy równanie (4) to (zamiast zwrócić error) mnożymy (4) i (5) stronami, uzyskując

$$w^k z^{2i} = 1 \pmod p.$$

Ponieważ lewa strona powyższej równości ma wartość

$$\left(w\left(z^{2i/k}\right)\right)^k,$$

to możemy kontynuować wykonanie algorytmu, po podstawieniu

$$w := w \cdot \left(z^{2i/k}\right). \tag{6}$$

Zauważmy (przyda nam się to za chwilę), że z konstrukcji algorytmu łatwo wynika, że $2i/k$ jest zawsze parzystą liczbą naturalną. Niewykluczone, że podstawienie (6) będziemy musieli wykonywać wielokrotnie, ale to nie szkodzi. Na koniec otrzymujemy w takie, że $w^{2j+1} = 1 \pmod p$. Co prawda w nie jest już równe x, ale wiemy, że $w = x \cdot z^{2l}$, gdzie l jest jakąś liczbą naturalną. Zatem $x^{2j+1} \cdot z^{2l(2j+1)} = 1 \pmod p$, z czego wynika, że $x^{2j+2} \cdot z^{2l(2j+1)} = x \pmod p$. Czyli $x^{j+1} \cdot z^{l(2j+1)}$ jest pierwiastkiem x modulo p.

Element $z \in \overline{\mathrm{QR}}_p$ jest znajdowany na chybił-trafił: bierzemy losowy element Z_p^* i sprawdzamy czy jest on resztą kwadratową (jeśli tak, to bierzemy następny i tak do skutku). Ponieważ szansa, że losowy element Z_p^* jest resztą kwadratową wynosi $1/2$, to oczekiwana liczba powtórzeń wynosi 2. Ten element algorytmu sprawia jednak, że jest on niedeterministyczny (choć oczekiwany czas działania jest wielomianowy). Nie jest znany wielomianowy algorytm deterministyczny dla tego problemu.

Powyższy algorytm zapisany w pseudo-kodzie można znaleźć np. w [MvOV97] (Algorytm 3.34). □

Z Chińskiego Twierdzenia o Resztach mamy teraz następujący fakt.

Wniosek 11 *Istnieje algorytm (działający w oczekiwanym czasie wielomianowym) pobierający na wejściu moduł RSA n oraz $x \in \mathrm{QR}_n$ i zwracający y takie, że $x = y^2 \pmod p$.*

9.2 Szyfrowanie

Generacja klucza Losujemy liczbę $n = pq$ w taki sam sposób jak w RSA (patrz Wykład 2, Rozdział 5.2). Kluczem publicznym jest n a kluczem prywatnym p i q. Zbiorem wiadomości jest Z_n^* a zbiorem szyfrogramów jest QR_n.

Funkcja szyfrująca

$$\mathcal{E}_n^{\mathrm{Rabin}}(x) := x^2 \bmod n$$

Funkcja odszyfrowująca

$$\mathcal{D}_{p,q}^{\mathrm{Rabin}}(y) := \sqrt{y} \bmod n,$$

gdzie $\sqrt{x} \bmod n$ jest zbiorem wszystkich pierwiastków y w Z_n^*.

Zauważmy, że taki kryptosystem nie spełnia warunku $x = \mathcal{D}_{p,q}^{\mathrm{Rabin}}(\mathcal{E}_n^{\mathrm{Rabin}}(x))$ (ale spełnia warunek $x \in \mathcal{D}_{p,q}^{\mathrm{Rabin}}(\mathcal{E}_n^{\mathrm{Rabin}}(x))$. Dlatego aby użyć kryptosystemu Rabina w praktyce, należy ograniczyć zbiór wiadomości np. przez ustalenie kodowania $\gamma : \{0,1\}^k \to Z_n^*$ (gdzie $2^k \ll |Z_n^*|$). (Takie kodowanie może np. polegać na dopisaniu ustalonej liczby jedynek na początku wiadomości.) Wówczas z dużym prawdopodobieństwem tylko jeden z elementów zbioru $\mathcal{D}_{p,q}^{\mathrm{Rabin}}(\mathcal{E}_n^{\mathrm{Rabin}}(x))$ jest możliwą wartością funkcji γ, więc odszyfrowanie jest jednoznaczne.

Twierdzenie 12 *Niech n i (p,q) będą (odpowiednio) losowym kluczem publicznym i prywatnym w kryptosystemie Rabina. Następujące stwierdzenia są równoważne:*

1. *istnieje WAP obliczający faktoryzację n*
2. *istnieje WAP obliczający (przynajmniej) jedną z wartości $\mathcal{D}_{p,q}^{\text{Rabin}}(y)$ dla $y = \mathcal{E}_n^{\text{Rabin}}(x)$, gdzie x jest losowe.*

Dowód. Natychmiastowa konsekwencja Lematu 9 i Wniosku 11. □

Wydawać by się mogło, że w związku z tym bezpieczeństwo kryptosystemu Rabina jest równoważne trudności faktoryzacji. Nie jest tak jednak, z następujących powodów.

- Po pierwsze: Punkt 2 mówi o trudności obliczenia całości wiadomości, a nie poszczególnych bitów.
- Po drugie: jak powiedzieliśmy, aby użyć kryptosystemu Rabina trzeba użyć jakiegoś kodowania γ, a wtedy tracimy równoważność z Twierdzenia 12.

  Jeśli nie użyjemy kodowania γ, albo jeśli $2^k / |Z_n^*|$ nie jest wystarczająco małe, to kryptosystem Rabina nie jest bezpieczny ze względu na atak z wybranym kryptogramem.

9.3 Schemat podpisu

Z szyfru Rabina możemy w standardowy sposób uzyskać schemat szyfrowania. W schemacie tym generacja pary kluczy jest identyczna jak w szyfrowaniu, przestrzenią wiadomości jest QR_n, a zbiorem podpisów jest Z_n^*. Podpisem na wiadomości x jest (dowolny) pierwiastek kwadratowy z x (t.j. y takie, że $y^2 = x \pmod n$). Aby zweryfikować podpis y wystarczy podnieść x do kwadratu (modulo n) i porównać z x. Umiejętność podpisania wybranej wiadomości x jest w tym przypadku obliczeniowo równoważna umiejętności faktoryzacji n (podobnie jak to miało miejsce w przypadku szyfrowania).

Taki schemat można całkowicie złamać za pomocą ataku z wybraną wiadomością. Wystarczy by przeciwnik uzyskał podpis y' na jakimś y^2 (dla losowego, znanego mu y). Ze szansą $1/2$ takie y' nie jest równe $\pm y \pmod n$ i (stosując metodę jak w dowodzie Lematu 9) przeciwnik może sfaktoryzować n.

Problem ten można obejść wprowadzając wymaganie, by podpisywane wiadomości miały ustalony format (podobnie jak zrobiliśmy to w Rozdziale 9.2). Taki zabieg powoduje jednak, że tracimy równoważność umiejętności podpisywania z umiejętnością faktoryzacji.

10 Notacja

QR_n zbiór reszty kwadratowych modulo n (patrz Rozdział 7),

$\overline{\text{QR}_n}$ zbiór elementów Z_n^*, które nie są resztami kwadratowymi modulo n (patrz Rozdział 7),

$\left(\frac{\cdot}{\cdot}\right)$ symbol Legendre'a (jeśli n jest pierwsza) albo symbol Jacobiego (patrz Rozdział 7).

Literatura

[Lip81] R. Lipton. How to cheat at mental poker. In *Proc. AMS Short Course on Cryptography*, 1981.

[MvOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997. dostępne pod adresem `http://www.cacr.math.uwaterloo.ca/hac/`.

[Rab79] M. Rabin. Digitalized signatures as intractable as factorization. Technical Report MIT/LCS/TR-212, MIT Laboratory for Computer Science, jan 1979.

[Sti02] Douglas Stinson. *Cryptography: theory and practice.* CRC Press, 2002.