Matematyka stojąca za RSA

Czyli właściwie dlaczego możemy czuć się bezpieczni

Mateusz Bielawski

Koło Naukowe Math4You
Wydział Informatyki Politechniki Białostockiej

mateusz.bielawski@long.int.pl

10 kwietnia 2017

Plan prezentacji

1. Podstawy teorii szyfrowania danych
   - Klucz asymetryczny
2. Matematyczne podstawy algorytmu RSA
   - Funkcja Eulera
   - Małe Twierdzenie Fermata
   - Chińskie twierdzenie o resztach
   - Kroki algorytmu generowania kluczy RSA
3. Bezpieczeństwo algorytmu
   - Problemy trudne obliczeniowo
   - Jak złamać RSA?
   - Problem faktoryzacji

Klucz asymetryczny

- Alice i Bob próbują się skomunikować.
- Każdy z nich ma po 2 klucze - jeden prywatny, utajniony; drugi publiczny.
- Wiadomość do Alice jest szyfrowana za pomocą jej klucza publicznego.
- Może ona odszyfrować tę wiadomość za pomocą swojego klucza prywatnego.
- Przykłady algorytmów: DSA, RSA, ECC.

Podpisanie wiadomości

- Alice potrzebuje wysłać wiadomość do Boba.
- Jednocześnie chce potwierdzić, że to na pewno ona wysłała tę wiadomość.
- Szyfruje ona swoją wiadomość za pomocą klucza publicznego Boba.
- Jednocześnie dodaje część wiadomości, którą szyfruje swoim kluczem prywatnym.
- Bob odszyfrowuje wiadomość za pomocą swojego klucza prywatnego.
- Kolejną część odszyfrowuje on kluczem publicznym Alice, potwierdzając tym sposobem autentyczność nadawcy.

Funkcja Eulera

Definicja

Funkcją Eulera $\varphi(n)$ nazywamy przyporządkowanie każdej liczbie naturalnej (bez zera) liczby liczb do niej względnie pierwszych, mniejszych lub równych tej liczbie.

n	1	2	3	4	5	6	7	8	9	10
$\varphi(n)$	1	1	2	2	4	2	6	4	6	4

Właściwości funkcji φ

- $\varphi(p) = p - 1$, gdy liczba p jest pierwsza

Właściwości funkcji φ

- $\varphi(p) = p - 1$, gdy liczba p jest pierwsza
- $\varphi(p \cdot q) = \varphi(p) \cdot \varphi(q)$, gdy p i q są względnie pierwsze

Małe Twierdzenie Fermata

Definicja

Dla liczb a i p, takich że a jest niezerową liczbą całkowitą i p jest liczbą pierwszą zachodzi równanie:

$$a^{p-1} \equiv_p 1$$

Chińskie twierdzenie o resztach

Definicja (dla przypadku z dwiema liczbami)

Niech p i q będą liczbami względnie pierwszymi. Liczba a, spełniająca warunek $a \equiv_p b$ i $a \equiv_q b$, spełnia także warunek $a \equiv_{pq} b$

Kroki algorytmu generowania kluczy RSA

- Wybieramy losowo dwie duże liczby pierwsze p i q.

Kroki algorytmu generowania kluczy RSA

- Wybieramy losowo dwie duże liczby pierwsze p i q.
- Liczymy wynik ich mnożenia $p \cdot q$, który nazwiemy n.

Kroki algorytmu generowania kluczy RSA

- Wybieramy losowo dwie duże liczby pierwsze p i q.
- Liczymy wynik ich mnożenia $p \cdot q$, który nazwiemy n.
- Wybieramy losowo taką liczbę naturalną e, która jest mniejsza od $(p-1)(q-1)$ i względnie z nią pierwsza.

Kroki algorytmu generowania kluczy RSA

- Wybieramy losowo dwie duże liczby pierwsze p i q.
- Liczymy wynik ich mnożenia $p \cdot q$, który nazwiemy n.
- Wybieramy losowo taką liczbę naturalną e, która jest mniejsza od $(p-1)(q-1)$ i względnie z nią pierwsza.
- Znajdujemy jej odwrotność w grupie modulo $(p-1)(q-1)$ ze zdefiniowaną operacją mnożenia - nazywamy to d

Kroki algorytmu generowania kluczy RSA

- Wybieramy losowo dwie duże liczby pierwsze p i q.
- Liczymy wynik ich mnożenia $p \cdot q$, który nazwiemy n.
- Wybieramy losowo taką liczbę naturalną e, która jest mniejsza od $(p-1)(q-1)$ i względnie z nią pierwsza.
- Znajdujemy jej odwrotność w grupie modulo $(p-1)(q-1)$ ze zdefiniowaną operacją mnożenia - nazywamy to d
- Parę liczb (e,n) nazywamy kluczem publicznym, natomiast (d,n) - prywatnym

Przykład

- Wybieramy dwie liczby pierwsze: 13 i 17

Przykład

- Wybieramy dwie liczby pierwsze: 13 i 17
- Mnożymy je przez siebie: 221

Przykład

- Wybieramy dwie liczby pierwsze: 13 i 17
- Mnożymy je przez siebie: 221
- Wybieramy liczbę względnie pierwszą z (13-1)(17-1) = 192: 101

Przykład

- Wybieramy dwie liczby pierwsze: 13 i 17
- Mnożymy je przez siebie: 221
- Wybieramy liczbę względnie pierwszą z (13-1)(17-1) $=$ 192: 101
- Znajdujemy element odwrotny: 173 ($101 \cdot 173 = 17473 \equiv_{192} 1$)

Przykład

- Wybieramy dwie liczby pierwsze: 13 i 17
- Mnożymy je przez siebie: 221
- Wybieramy liczbę względnie pierwszą z (13-1)(17-1) $= 192$: 101
- Znajdujemy element odwrotny: 173 ($101 \cdot 173 = 17473 \equiv_{192} 1$)
- Para liczb (101,221) jest kluczem publicznym, (173,221) - prywatnym

Jak to działa i dlaczego?

Twierdzenie

$$m^{ed} \equiv_n m$$

Jak to działa i dlaczego?

Twierdzenie

$$m^{ed} \equiv_n m$$

Dowód

- $ed \equiv \varphi(n) 1$
- $n = pq$
- $ed = 1 + k(q-1) = 1 + h(p-1)$
- $m^{ed} \equiv_p m^{1+h(p-1)} \equiv_p m(m^{p-1})^h \equiv_p m(1)^h \equiv_p m$
- $m^{ed} \equiv_q m^{1+k(q-1)} \equiv_q m(m^{q-1})^k \equiv_q m(1)^k \equiv_q m$
- Na mocy chińskiego twierdzenia o resztach $m^{ed} \equiv_n m$ Q.E.D.

Problemy trudne obliczeniowo

W algorytmice problemami trudnymi obliczeniowo nazywamy takie problemy, dla których nie istnieje (bądź nie został jeszcze odkryty) algorytm poprawny (to znaczy taki, który zawsze podaje poprawną wartość) o złożoności wielomianowej, tj. takie, dla których liczba operacji rośnie proporcjonalnie do jakiegoś wielomianu.
Przykłady:

- Problem faktoryzacji
- Problem komiwojażera
- Problem znalezienia cyklu Hamiltona w grafie
- Problem kolorowania grafu

Jak złamać RSA?

- Mamy do dyspozycji (e,n) - klucz publiczny uzytkownika

Jak złamać RSA?

- Mamy do dyspozycji (e,n) - klucz publiczny uzytkownika
- Aby móc rozszyfrować/szyfrować wiadomości jak ten użytkownik musimy mieć d - jego klucz prywatny

Jak złamać RSA?

- Mamy do dyspozycji (e,n) - klucz publiczny uzytkownika
- Aby móc rozszyfrować/szyfrować wiadomości jak ten użytkownik musimy mieć d - jego klucz prywatny
- Wiemy, że d i e spełniają równanie $de \equiv_{\varphi(n)} 1$.

Jak złamać RSA?

- Mamy do dyspozycji (e,n) - klucz publiczny uzytkownika
- Aby móc rozszyfrować/szyfrować wiadomości jak ten użytkownik musimy mieć d - jego klucz prywatny
- Wiemy, że d i e spełniają równanie $de \equiv_{\varphi(n)} 1$.
- n jest iloczynem dwóch liczb pierwszych p i q, ergo: trzeba rozłożyć ją na czynniki pierwsze, aby poznać p i q.

Jak złamać RSA?

- Mamy do dyspozycji (e,n) - klucz publiczny uzytkownika
- Aby móc rozszyfrować/szyfrować wiadomości jak ten użytkownik musimy mieć d - jego klucz prywatny
- Wiemy, że d i e spełniają równanie $de \equiv_{\varphi(n)} 1$.
- n jest iloczynem dwóch liczb pierwszych p i q, ergo: trzeba rozłożyć ją na czynniki pierwsze, aby poznać p i q.
- Kiedy już znamy p i q to wyliczenie klucza prywatnego to problem prosty, rozwiązywalny poprzez rozszerzony algorytm Euklidesa.

Problem faktoryzacji

- Do tej pory nie udowodniono, że da się znaleźć poprawny algorytm wielomianowy do tego problemu.
- Na tym przypuszczeniu opiera się bezpieczeństwo RSA.
- Podstawy bezpieczeństwa RSA są zagrożone przez rozwijające się komputery kwantowe, na których faktoryzacja przebiega szybciej $O(\log^3(n))$.