Algorytmy i struktury danych

Autor artykułu: [mgr Jerzy Wałaszek](mailto:i-lo@eduinf.waw.pl?subject=Struktury%20danych)

Symetryczny system szyfrowania to taki, w którym klucz szyfrujący pozwala zarówno szyfrować dane, jak również odszyfrowywać je. Opisane w poprzednich rozdziałach systemy były systemami symetrycznymi. Podstawową wadą systemów symetrycznych jest ścisła konieczność ochrony klucza. Z tego powodu można je było stosować tylko w ograniczonych grupach użytkowników.



W roku 1977 trzej profesorowie z MIT w USA, Ronald L. Rivest, Adi Shamir i Leonard Adleman, opublikowali nowy rodzaj szyfrowania danych, który nazwano od pierwszych liter ich nazwisk systemem RSA. Jest to niesymetryczny algorytm szyfrujący, którego zasadniczą cechą są dwa klucze: publiczny do kodowania informacji oraz prywatny do jej odczytywania. Klucz publiczny (można go udostępniać wszystkim zainteresowanym) umożliwia jedynie zaszyfrowanie danych i w żaden sposób nie ułatwia ich odczytania, nie musi więc być chroniony. Dzięki temu firmy dokonujące transakcji poprzez sieć Internet mogą zapewnić swoim klientom poufność i bezpieczeństwo. Drugi klucz (prywatny, przechowywany pod nadzorem) służy do odczytywania informacji zakodowanych przy pomocy pierwszego klucza. Klucz ten nie jest udostępniany publicznie. System RSA umożliwia bezpieczne przesyłanie danych w środowisku, w którym może dochodzić do różnych nadużyć. Bezpieczeństwo oparte jest na trudności rozkładu dużych liczb na czynniki pierwsze.

**Przykład:**

Załóżmy, iż dysponujemy superszybkim komputerem, który jest w stanie sprawdzić podzielność miliarda dużych liczb w ciągu jednej sekundy. Aby złamać szyfr **RSA** należy rozbić klucz publiczny na dwie liczby pierwsze będące jego dzielnikami. Znajomość tych liczb pozwala rozszyfrować każdą informację zakodowaną kluczem prywatnym i publicznym.

Brzmi dosyć prosto. Jednakże nie ma prostej metody rozbijania dużych liczb na czynniki pierwsze. Nie istnieje żaden wzór, do którego podstawiamy daną liczbę i w wyniku otrzymujemy wartości jej czynników pierwszych. Należy je znaleźć testując podzielność kolejnych liczb.

Z rozważań o liczbach pierwszych wynika, iż w przypadku dwóch różnych dzielników pierwszych jeden musi leżeć poniżej wartości pierwiastka z danej liczby, a drugi powyżej (dlaczego?). Zatem, aby go znaleźć musimy wyliczyć pierwiastek z rozkładanej liczby, a następnie testować podzielność przez liczby nieparzyste leżące poniżej tego pierwiastka.

Statystycznie poszukiwany czynnik pierwszy powinien znajdować się w górnej połówce zakresu od 2 do pierwiastka z n. Ile działań musimy wykonać? Policzmy.

Klucz 128 bitowy. Pierwiastek jest liczbą 64 bitową. W zakresie od 2 do 264 co druga liczba jest nieparzysta, zatem jest ich około 264 / 2 = 263. Ponieważ interesuje nas tylko górna połówka, to ilość liczb do sprawdzenia jest dwa razy mniejsza, czyli wynosi 263 / 2 = 262. Ile czasu zajmie naszemu superkomputerowi sprawdzenie podzielności przez około 262 liczb, jeśli w ciągu 1 sekundy wykonuje on miliard sprawdzeń? Odpowiedź brzmi: zajmie to około:

262/109 = 4611686018 sekund = 76861433 minut = 1281023 godzin = 53375 dni = 146 lat

Czy sądzisz, że ktoś będzie czekał przez prawie dwa życia na złamanie szyfru? Zatem można podać do publicznej wiadomości liczbę będącą iloczynem dwóch dużych liczb pierwszych i mieć prawie pewność, iż nikt jej nie rozbije na czynniki pierwsze w rozsądnym czasie. Ostatecznie zamiast 128 bitów możemy zwiększyć klucz do np. 1024 bitów, a wtedy czas łamania szyfru liczy się miliardami miliardów… miliardów lat.

**Fazy algorytmu RSA**

Algorytm **RSA** składa się z trzech podstawowych kroków:

1. Generacja **klucza publicznego** i **tajnego**. **Klucz publiczny** jest przekazywany wszystkim zainteresowanym i umożliwia zaszyfrowanie danych. **Klucz tajny** umożliwia rozszyfrowanie danych zakodowanych **kluczem publicznym**. Jest przechowywany on bezpiecznie bez możliwości udostępniania.
2. Użytkownik po otrzymaniu **klucza publicznego**, np. poprzez sieć Internet, koduje za jego pomocą swoje dane i przesyła je w postaci szyfru RSA do  adresata dysponującego **kluczem tajnym**, np. do banku, firmy komercyjnej, tajnych służb. **Klucz publiczny** nie musi być chroniony, ponieważ nie umożliwia on rozszyfrowania informacji – proces szyfrowania nie jest odwracalny przy pomocy tego klucza. Zatem nie ma potrzeby jego ochrony i  może on być powierzany wszystkim zainteresowanym bez ryzyka złamania kodu.
3. Adresat po otrzymaniu zaszyfrowanej wiadomości rozszyfrowuje ją za pomocą **klucza tajnego**.

## Tworzenie kluczy RSA

|  |  |
| --- | --- |
| **I** | Znajdź dwie duże liczby pierwsze (mające np. po 128 bitów). Oznacz je jako **p** i **q**. Istnieją specjalne algorytmy generujące duże liczby pierwsze, które wykorzystują np. [test Millera-Rabina](https://eduinf.waw.pl/inf/alg/001_search/0019.php). |
| **II** | Oblicz:  **Ø** = (**p**-1)×(**q**-1)  oraz  **n** = **p**×**q**  Liczby pierwsze **p** i **q** usuń, aby nie wpadły w niepowołane ręce. **Ø** to tzw. funkcja Eulera, **n** jest modułem. |
| **III** | Wykorzystując odpowiednio [algorytm Euklidesa](https://eduinf.waw.pl/inf/alg/001_search/0006.php) znajdź liczbę **e**, która jest względnie pierwsza z wyliczoną wartością funkcji Eulera **Ø** (tzn. NWD(e, Ø) = 1) Liczba ta powinna również spełniać nierówność 1 < **e** < **n**. Nie musi być pierwsza lecz nieparzysta. |
| **IV** | Oblicz liczbę odwrotną modulo **Ø** do liczby **e**, czyli spełniającą równanie:  **d**×**e** **mod** **Ø** = 1  Można to zrobić przy pomocy [rozszerzonego algorytmu Euklidesa](https://eduinf.waw.pl/inf/alg/001_search/0009.php), który umieściliśmy w naszym artykule. |
| **V** | Klucz publiczny jest parą liczb (**e**, **n**), gdzie **e** nazywa się publicznym wykładnikiem. Możesz go przekazywać wszystkim zainteresowanym. |
| **VI** | Klucz tajny to (**d**, **n**), gdzie **d** nazywa się prywatnym wykładnikiem. Klucz ten należy przechowywać pod ścisłym nadzorem. |

**Przykład:**

|  |  |
| --- | --- |
| **p = 13 q = 11** | Wybieramy dwie dowolne liczby pierwsze. W naszym przykładzie nie będą one duże, aby nie utrudniać obliczeń. W rzeczywistości liczby te powinny być ogromne. |
| **Ø = 120** | Obliczamy:  **Ø** = (**p**-1)×(**q**-1)  czyli tzw. funkcję Eulera:  **Ø** = (13-1)×(11-1)  = 12×10  = **120** |
| **n = 143** | Obliczamy moduł **n**:  **n** = **p**×**q**  = 13×11  = **143** |
| **e = 7** | Wyznaczamy wykładnik publiczny **e**. Ma on być względnie pierwszy z **Ø** czyli z liczbą 120. Warunek ten spełnia, np. liczba **7**. |
| **d = 103** | Wyznaczamy następnie wykładnik prywatny, który ma być odwrotnością modulo **Ø** liczby **e**, czyli:  **d**×7 **mod** 120 = 1  Liczbą spełniającą ten warunek jest **103**. |
| **(7, 143)** | Klucz publiczny (**e**, **n**). |
| **(103, 143)** | Klucz tajny (**d**, **n**). |

## Szyfrowanie kluczem publicznym RSA

|  |  |
| --- | --- |
| **I** | Otrzymujesz od adresata klucz publiczny w postaci pary liczb (**e**, **0**). |
| **II** | Wiadomość do zaszyfrowania zamieniasz na liczby naturalne **t**, które muszą spełniać nierówność  0 < **t** < **n**  Można tutaj skorzystać np. z łączenia kodów znaków. Oczywiście adresat musi znać użyty przez ciebie sposób przekształcenia tekstu w liczbę, aby mógł on później odtworzyć otrzymaną wiadomość. Zwykle nie ma z tym problemu, ponieważ nadawca i odbiorca stosują wspólne oprogramowanie, które troszczy się za ciebie o takie szczegóły techniczne. |
| **III** | Na tak otrzymanych liczbach wykonujesz operację szyfrowania i otrzymujesz liczby  **c** = **te** **mod** **n**. |
| **IV** | Liczby **c** są zaszyfrowaną postacią liczb **t** i przekazuje się je adresatowi wiadomości. Klucz (**e**, **n**) umożliwił ich zaszyfrowanie, lecz nie pozwala ich rozszyfrować. |

**Przykład:**

|  |  |
| --- | --- |
| **e = 7 n = 143** | Otrzymaliśmy klucz publiczny (**e**, **n**). Przy jego pomocy możemy zakodować liczby od 0 do 142. Zauważ, iż liczby 0 oraz 1 nie zostaną zakodowane (dlaczego?). |
| **c = 7** | Załóżmy, iż chcemy przesłać adresatowi zaszyfrowaną liczbę **t** = 123. W tym celu musimy obliczyć wartość wyrażenia:  **c** = 1237 **mod** 143  = 425927596977747 **mod** 143  = **7**  Wynik jest zaszyfrowaną liczbą **123**. Przesyłamy go do adresata. |

## Rozszyfrowywanie kluczem prywatnym RSA

|  |  |
| --- | --- |
| **I** | Jesteś adresatem zaszyfrowanych wiadomości. Wcześniej wszystkim korespondentom przesłałeś wygenerowany klucz publiczny (**e**, **n**), za pomocą którego mogą oni szyfrować i przesyłać ci swoje dane. Otrzymujesz więc zaszyfrowaną wiadomość w postaci liczb naturalnych **c**, które muszą spełniać warunek:  0 < **c** < **n** |
| **II** | Liczbę **c** przekształcasz na pierwotną wartość **t**, stosując wzór:  **t** = **cd** **mod** **n** |
| **III** | Z otrzymanej liczby **t** odtwarzasz wg ustalonego systemu znaki tekstu. Teraz możesz odczytać przesłaną wiadomość. |

**Przykład:**

|  |  |
| --- | --- |
| **d = 103 n = 143 c = 7** | Otrzymaliśmy zakodowaną wiadomość o wartości 7. Jesteśmy w  posiadaniu klucza prywatnego, który służy do rozszyfrowywania wiadomości zakodowanych kluczem publicznym. |
| **t = 123** | Wykonujemy następujące operacje:  **t** = **7103 mod 143**  Potęga jest zbyt duża, aby można ją było w normalny sposób obliczyć (języki programowania mają zwykle ograniczenia co do wielkości liczb całkowitych, wyjątkiem jest tutaj Python). Jednakże nas nie interesuje wartość liczbowa potęgi, a jedynie reszta z dzielenia jej przez **143**. Możemy więc rozłożyć potęgę na iloczyn składników o  wykładnikach równych kolejnym potęgom liczby dwa:  7103 **mod** 143 = 764+32+4+2+1 **mod** 143  = (764 **mod** 143)×(732 **mod** 143)×(74 **mod** 143)×(72 **mod** 143)×7 **mod** 143  71  **mod** 143 = 7 72  **mod** 143 = (71  **mod** 143)2 **mod** 143 = 49**mod** 143 = 49  74  **mod** 143 = (72  **mod** 143)2 **mod** 143 = 492 **mod** 143 = 113 78  **mod** 143 = (74  **mod** 143)2 **mod** 143 = 1132 **mod** 143 = 42 716 **mod** 143 = (78  **mod** 143)2 **mod** 143 = 422 **mod** 143 = 48 732 **mod** 143 = (716 **mod** 143)2 **mod** 143 = 482 **mod** 143 = 16 764 **mod** 143 = (732 **mod** 143)2 **mod** 143 = 162 **mod** 143 = 113  Do wyliczenia potęgi bierzemy tylko te reszty, które występują w sumie potęg 2: (jeśli byłoby ich bardzo dużo, to każde kolejne mnożenie można wykonać z operacją modulo, dzięki czemu wynik nigdy nie wyjdzie poza wartość modułu)  **t** = 7103 **mod** 143  = 113×16×113×49×7 **mod** 143  = 123 |

## Przykładowe zastosowania RSA

### Bezpieczne połączenie internetowe

Sieć komputerowa Internet jest środowiskiem o niskim bezpieczeństwie poufności przesyłanych danych. Pakiety danych podróżujące pomiędzy różnymi węzłami sieci mogą być podglądane przez osoby nieupoważnione. Szyfrowanie danych zapewni nam bezpieczeństwo. Nawiązanie bezpiecznego połączenia wykorzystującego szyfrowanie RSA składa się z następujących etapów:

1. Obie stacje generują zestaw **kluczy RSA**.
2. Stacje wymieniają się **kluczami publicznymi**, które posłużą do szyfrowania przesyłanych wiadomości. Operacja ta jest bezpieczna, ponieważ klucze publiczne nie pozwalają odczytać zaszyfrowanych przy ich pomocy wiadomości. Zatem przechwycenie klucza publicznego przez osobę nieupoważnioną nie da jej żadnych korzyści.
3. Wysyłane wiadomości stacje szyfrują przy pomocy otrzymanego **klucza publicznego**.
4. Odebrane wiadomości stacje rozszyfrowują przy pomocy swojego **klucza prywatnego**, który nie był ujawniany. Dzięki temu przechwycenie szyfrogramu w drodze do odbiorcy nie przyniesie osobie nieupoważnionej żadnych korzyści.

Bezpieczne połączenia internetowe są dzisiaj szeroko wykorzystywane w sieci do prowadzenia działalności handlowej. Dzięki nim klienci banków mogą bezpiecznie zarządzać swoimi kontami oraz dokonywać zakupów w sieci z wykorzystaniem kart płatniczych.

### Podpis cyfrowy

Załóżmy, iż stacja A chce wysłać do  stacji B wiadomość **W** podpisaną cyfrowo.

1. W tym celu stacja A szyfruje wiadomość **W** za pomocą swojego **klucza tajnego** i szyfr dołącza do tej wiadomości. **Klucz tajny** stacja A otrzymuje od  instytucji zajmującej się przydzielaniem certyfikatów – jest to tzw. **podpis elektroniczny**, który jednoznacznie identyfikuje nadawcę wiadomości. W efekcie nowa wiadomość **W'** składa się z oryginalnej wiadomości **W** oraz jej zaszyfrowanej kopii.
2. W takiej postaci wiadomość **W'** zostaje przesłana do stacji B.
3. Stacja B i potwierdzony przez instytucję przydzielającą certyfikaty. W ten sposób stacja B ma pewność, iż **klucz publiczny** na pewno dotyczy stacji A.
4. Stacja B porównuje obie części wiadomości **W'**. Jeśli są takie same, to oznacza to, iż pochodzą rzeczywiście od stacji A.

Jeśli przesyłana wiadomość jest poufna, to do jej przekazania można dodatkowo wykorzystać bezpieczne połączenie internetowe.