3. Szyfry z kluczem publicznym

3.1 Wprowadzenie

Krótki, zwarty opis systemów kryptograficznych z kluczem publicznym (m.in. szyfru RSA, szyfru Rabina, szyfru ElGamala i szyfrów plecakowych) można znaleźć między innymi w książkach Menezesa [1], Schneiera [6], Stokłosy [2], Koblitza [3],[4] i Kutyłowskiego [5]. Przejrzyste, staranne wprowadzenie do systemów z kluczem publicznym można również znaleźć w podręczniku T.HCormen,'a C.E.Leiserson'a i R.L.Rivest'a „Wprowadzenie do algorytmów" [7].

Systemy kryptograficzne z kluczem publicznym są systemami z kluczem asymetrycznym.

Warto zwrócić uwagę na fakt, że określenia „kryptografia z kluczem publicznym" czy „algorytmy kryptografii z kluczem publicznym" dotyczą nie tylko systemów kryptograficznych czyli szyfrów ale również podpisów cyfrowych, uwierzytelniania i dystrybucji kluczy.

Ogólnie rzecz biorąc kryptografii z kluczem publicznym PK (ang. public key cryptography) można używać do:

1. szyfrowania (np. szyfr RSA, szyfr Rabina, szyfr ElGamala, szyfry plecakowe, szyfr Massey'a-Omury)

2. podpisów cyfrowych

3. uwierzytelniania

4. dystrybucji kluczy prywatnych dla celów kryptografii z kluczem prywatnym (tak jest np. w programie do szyfrowania PGP)

W tym rozdziale zajmujemy się jednak tylko algorytmami szyfrowania z kluczem publicznym.

Zamiana tekstu jawnego na liczbę lub ogólniej element struktury algebraicznej w której prowadzimy obliczenia

Wszystkie szyfry z kluczem publicznym są szyframi blokowymi. Długość bloku $l_1 \in N$ dobieramy do ilości elementów q pierścienia lub ciała w którym prowadzimy obliczenia.

Załóżmy, dla ustalenia uwagi, że prowadzimy obliczenia w ciele $F_q = \{0,1,...,q-1\}$, gdzie q jest liczbą pierwszą. Dobieramy długość bloku $l_1 \in N$ tak by $N_0^{l_1} < q$, gdzie $N_0 = card(V_1)$ i V_1 jest alfabetem w którym zapisujemy wiadomości jawne. Wszystkim blokom czyli jednostkom tekstu tzn. wszystkim słowom o długości l_1 nad alfabetem V_1

przyporządkowujemy różnowartościowo elementy z ciała skończonego F_q w dowolny sposób (ponieważ $N_0^{l_1} < q$ to istotnie możemy tak zrobić).

Na przykład możemy, co jest bardzo wygodne, traktować słowa z $V_1^{l_1}$ jako słowa kodowe reprezentujące liczby z ciała F_q w naturalnym zapisie wagowym z wagą $N_0 = card(V_1)$ czyli naszym odwzorowaniem zamieniającym tekst na liczbę jest odwzorowanie różnowartościowe:

$$V_1^{l_1} \ni m = m_1 m_2 ... m_n \rightarrow x = m_1 N_0^{l_1 - 1} + m_2 N_0^{l_2 - 2} + ... + m_{n-1} N_0 + m_n \in N \cup \{0\}$$

Mamy więc metodę pozwalającą utożsamić tekst i liczbę. Opisując algorytm szyfrowania z kluczem publicznym wręcz mówimy np. „niech liczba $m \in F_q$ będzie wiadomością jawną".

Można by też postąpić inaczej. Jeśli mamy litery $m_1, m_2, ..., m_n$ tekstu jawnego $m = m_1 m_2 ... m_n$ zakodowane 8 bitowym binarnym kodem ASCII to można potraktować blok tekstu będący w tej sytuacji słowem binarnym jako liczbę w zapisie NKB. Mamy więc inną metodę pozwalającą utożsamić tekst i liczbę.

3. 2 System kryptograficzny RSA

1. Szyfr *RSA* jest typowym, najczęściej stosowanym w praktyce szyfrem z kluczem publicznym. Szyfr *RSA* jest szyfrem asymetrycznym jak wszystkie szyfry z kluczem publicznym. Nazwa szyfru *RSA* pochodzi od nazwisk trzech matematyków, profesorów z MIT (Massachussets Institute of Technology) Rona L.Rivesta, Adi Shamira i Leonarda M.Adlemana, którzy w roku 1977 opracowali koncepcję tego szyfru. Szyfr RSA uważany jest za jeden z najbezpieczniejszych, najpewniejszych szyfrów. Używany jest do szyfrowania wiadomości, podpisów cyfrowych oraz dystrybucji kluczy kryptograficznych.

Rivest, Shamir i Adleman w słynnej zagadce zamieszczonej w 1977 roku w czasopiśmie "Scientific American" zamieścili szyfrogram pewnego tekstu jawnego, proponując czytelnikom odtworzenie tego tekstu z szyfrogramu. Użyli przy tym szyfru nazwanego później *RSA*-129. Liczba 129 w oznaczeniu bierze się od 129 cyfrowej liczby (cyfry dziesiętne), którą trzeba rozłożyć na czynniki pierwsze by uzyskać klucz deszyfrujący i funkcję odwrotną deszyfrującą szyfrogram. Zadanie przed którym stanęli czytelnicy było więc typowym problemem przed którym staje kryptoanalityk. Szyfr *RSA*-129 został w końcu złamany w 1994 roku dzięki równoległej pracy wielu komputerów w sieci komputerowej i superkomputerów równoległych. Inny szyfr *RSA* o nazwie RSA-150 nie został jeszcze do dzisiaj złamany.

Algorytmy *RSA* mające klucz o długości 512 bitów (lub mniejszej), są stosunkowo proste do złamania. Adi Shamir opracował w 1999 r. specjalizowany równoległy komputer łamiący 512 bitowe RSA w ciągu 2 dni. Obecnie stosuje się klucze o długości 1024 bitów, 2048 bitów (długość zalecana przez amerykańską firmę kryptograficzną RSA) a nawet klucze 4096 bitowe praktycznie całkowicie bezpieczne tzn. nie do złamania przy współczesnym stanie wiedzy o algorytmach komputerowych i przy współczesnych możliwościach obliczeniowych systemów cyfrowych. Uważa się, że klucze o długości 1024 bitów będą całkowicie bezpieczne do 2010 roku. Zatem typowa długość klucza w RSA to 1024 do 4096 bitów.

Algorytm *RSA* został przez autorów opatentowany. Patent opiewał na Massachussets Institute of Technology i obejmował zarówno szyfrowanie jak i podpisy cyfrowe. Obecnie ważność patentu już wygasła.

2. Wiadomości jawne i wiadomości tajne (czyli jak mówimy szyfrogramy lub kryptogramy) utożsamiane są w RSA z liczbami z pierścienia, $Z_n = \{0,1,...,n-1\}$, gdzie $n = p_1 p_2$ i p_1, p_2 są różnymi liczbami pierwszymi. Przedstawimy poniżej szyfr RSA w wersji nieco ogólniejszej tzn. dla pierścienia Z_n, dla którego $n = p_1 p_2 ... p_k$ i $p_1, p_2, ..., p_k$ są parami różnymi liczbami pierwszymi czyli $p_i \neq p_j$ dla $i \neq j$. Poniżej opisane są krótko podstawy matematyczne algorytmu RSA w tej ogólniejszej wersji.

Wybieramy dwie duże liczby pierwsze p i q obliczamy iloczyn $n=pq$ oraz wartość funkcji Eulera $\varphi(n) = (p-1)(q-1)$. Wybieramy następnie losowo dowolną taką liczbę $d \in Z_{\varphi(n)}$, dla której istnieje element odwrotny $e = d^{-1}$ w pierścieniu $Z_{\varphi(n)}$ czyli takie $e \in Z_{\varphi(n)}$ by $d \otimes_{\varphi(n)} e = 1$ w pierścieniu $Z_{\varphi(n)}$ (oczywiście jest to równoważne temu by znaleźć takie $e \in Z_{\varphi(n)}$, żeby $d \cdot e \equiv 1 \pmod{\varphi(n)}$.) Warunkiem koniecznym i dostatecznym na to by istniał taki element odwrotny jest by $NWD(\varphi(n), d) = 1$. Obliczamy $e = d^{-1}$ i ujawniamy e jako klucz

publiczny. Dokładniej kluczem publicznym jest para uporządkowana (*e*,*n*). Kluczem prywatnym będzie *d* a dokładniej para uporządkowana (*d*,*n*).

1. Tekst szyfrowany, czyli tekst jawny $m \in V_1^*$ np. "ala ma kota" zamieniamy na liczbę całkowitą *x* np. 10982398.

2. Podnosimy tę liczbę do ustalonej potęgi *e* w pierścieniu Z_n *((e,n)* jest znane i stanowi klucz publiczny) lub co na jedno wychodzi podnosimy tę liczbę do potęgi e w pierścieniu Z i bierzemy resztę z dzielenia tej liczby przez liczbę *n.*Tę resztę wysyłamy do adresata.

$$c = x^e$$

3. Adresat podnosi (w pierścieniu Z_n) uzyskaną liczbę do pewnej potęgi d będącej kluczem tajnym czyli kluczem prywatnym uzyskując wiadomość jawną x.

$$x = c^d$$

Twierdzenie (lemat do twierdzenia o poprawności RSA)
Niech $m_1, m_2, ..., m_n \in N$ będą parami względnie pierwsze (tzn. $NWD(m_i, m_j) = 1$ dla każdego $i, j \in <1, n>,\ i \neq j$) i $m_i \geq 2$ dla każdego i=1,2,...,*n* oraz niech *a, b* będą dowolnymi liczbami całkowitymi czyli $a, b \in Z$.

$a \equiv b(\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,*n* wt. i tylko wt. gdy $a \equiv b(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$ (8.1)

Dowód. Załóżmy, że $a \equiv b(\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,n. Układ kongruencji $a \equiv b(\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,n możemy zapisać równoważnie jako $a - b \equiv 0(\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,n lub oznaczając $x \overset{ozn}{=} a - b$ w postaci $x \equiv 0(\mathrm{mod}\, m_i)$ dla każdego *i*=1,2,...,*n*.

Z chińskiego twierdzenia o resztach układ kongruencji $x \equiv 0(\mathrm{mod}\, m_i)$ dla każdego *i*=1,2,...,*n* ma w zbiorze $< 0, m_1 \cdot m_2 \cdot ... \cdot m_n - 1 >$ dokładnie jedno rozwiązanie i jak łatwo sprawdzić jest nim x=0. Jednocześnie ogólne rozwiązanie tego układu kongruencji dane jest wzorem $x = k \cdot m_1 \cdot m_2 \cdot ... \cdot m_n$, gdzie $k \in Z$. Zatem $x \equiv 0\,(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$ co oznacza, że $a - b \equiv 0\,(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$. Z własności kongruencji wynika więc ostatecznie, że $a \equiv b(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$ co dowodzi wynikania w prawo.

Odwrotnie załóżmy, że $a \equiv b(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$ i ustalmy m_i wówczas istnieje takie $k \in Z$, że $a = b + k \cdot m_1 \cdot m_2 \cdot ... \cdot m_n$ zatem (z własności homomorfizmu $[\cdot]_{m_i} : Z \to Z_{m_i}$ jakim jest

branie reszty modulo m_i) mamy $a \equiv b(\mathrm{mod}\, m_i)$ co z uwagi na dowolność m_i dowodzi wynikania w lewo. ■

Uwaga 1. Założenie, że $m_1, m_2, ..., m_n \in N$ są parami względnie pierwsze jest istotne bo np. dla $m_1 = 2$, m_2=6 mamy dla $a = 7$, $b = 1$: $7 \equiv 1(\mathrm{mod}\, 6)$, $7 \equiv 1(\mathrm{mod}\, 2)$ ale 7 nie przystaje do 1 modulo 12.

Uwaga 2.Do wynikania w lewo założenie, że $m_1, m_2, ..., m_n \in N$ są parami względnie pierwsze nie jest potrzebne tzn. jeśli $a \equiv b(\mathrm{mod}\, m_1 \cdot m_2 \cdot ... \cdot m_n)$ to $a \equiv b(\mathrm{mod}\, m_i)$ dla każdego i=1,2,...,*n*.

Twierdzenie (o poprawności szyfru RSA)
Załóżmy że liczba $n \in N$ jest bezkwadratowa (tzn. $n = pq$, gdzie p, q są parami różnymi liczbami pierwszymi) oraz niech $m \in Z_n$. Niech ponadto dwie liczby naturalne $d, e \in Z_{\varphi(n)}$ (gdzie φ jest funkcją Eulera) będą takie, że $d \otimes_{\varphi(n)} e = 1$ czyli *d* i *e* są względem siebie odwrotne w pierścieniu $Z_{\varphi(n)} = \{0,1,2,...,\varphi(n)-1\}$ (lub równoważnie $d \cdot e \equiv 1 \pmod{\varphi(n)}$. Wówczas

$$(m^d)^e \equiv m \pmod{n}$$

co (jeśli podnoszenie do potęgi traktujemy jako podnoszenie do potęgi w Z_n) można zapisać jako $(m^d)^e = m$.

Uwaga 1. Jest to zasadnicze twierdzenie dla poprawności szyfru RSA. Mówi ono bowiem, że z kryptogramu $c = m^d$ możemy odzyskać wiadomość jawną *m* podnosząc kryptogram $c = m^d$ do potęgi *e*.

Dokładniej użyteczność powyższego twierdzenia dla kryptografii polega na tym, że przy pewnych warunkach narzuconych na liczby *n, d, e*, funkcje potęgowe w pierścieniu Z_n zdefiniowane jako $Z_n \ni x \to x^d \in Z_n$ oraz $Z_n \ni x \to x^e \in Z_n$ są względem siebie odwrotne.

Uwaga 2. Warto zwrócić uwagę na to, że nie wymagamy w założeniach powyższego twierdzenia by $NWD(m,n) = 1$ tak jak w założeniach twierdzenia Eulera (por. rozdz. 1).

Uwaga 3. Dla danego *e* odwracalnego istnieje tylko jedno *d* takie, że $d \otimes_{\varphi(n)} e = 1$. Element odwrotny w pierścieniu przemiennym z 1 wyznaczony jest bowiem jednoznacznie. Czyli do danego klucza publicznego pasuje tylko jeden klucz prywatny i odrotnie

Twierdzenie (o poprawności uogólnionej wersji szyfru RSA)
Załóżmy że liczba $n \in N$ jest bezkwadratowa tzn. $n = p_1 p_2 p_3 ... p_k$, gdzie $p_1, p_2, p_3, ..., p_k$ są parami różnymi liczbami pierwszymi oraz niech $m \in Z_n$. Niech dwie liczby naturalne $d, e \in N$ będą takie, że $d \otimes_{\varphi(n)} e = 1$ (lub równoważnie $d \cdot e \equiv 1 \pmod{\varphi(n)}$. Wówczas

$$(m^d)^e \equiv m \pmod{n}$$

lub równoważnie $a^{de} = a$ w pierścieniu Z_n (potęgujemy w pierścieniu Z_n).

Dowód. 1. Dowód przeprowadzimy dla liczby bezkwadratowej $n = pq$, dla liczby bezkwadratowej $n = p_1 p_2 ... p_k$ dowód jest analogiczny.

2. Dla $m = 0$ kongruencja z tezy twierdzenia jest oczywista. Będziemy więc zakładać w dalszym ciągu, że $m \neq 0$

3. Z założenia $ed \equiv 1 \pmod{\varphi(n)}$ zatem istnieje takie $k \in Z$, że

$$ed = 1 + k \cdot \varphi(n) \qquad (*)$$

Jednocześnie ponieważ $\varphi(n) = (p-1)(q-1)$ dostajemy, że $ed = 1 + k \cdot (p-1)(q-1)$

4. Załóżmy teraz, że wiadomość jawna $m \in Z_n$ jest względnie pierwsza z p, czyli $NWD(m, p) = 1$. Z małego twierdzenia Fermata mamy wówczas:

$$m^{p-1} \equiv 1 \pmod{p} \qquad (**)$$

Podnosząc obie strony kongruencji (**) do potęgi $k \cdot (q-1)$ i mnożąc przez *m* dostajemy:

$$m^{1+k(p-1)(q-1)} \equiv m \pmod{p}$$

a więc korzystając z (*) mamy:

$$m^{ed} \equiv m \pmod{p}.$$

5. Jeżeli $NWD(m, p) > 1$, to oczywiście $NWD(m, p) = p$ (wtej sytuacji m jest wielokrotnością p). Zatem również mamy wówczas:

$$m^{ed} \equiv m(\mathrm{mod}\, p) \qquad (***)$$

Istotnie, ponieważ liczba m jest podzielna przez p, więc $m^{ed} \equiv 0(\mathrm{mod}\, p)$ oraz $m \equiv 0(\mathrm{mod}\, p)$ i kongruencja (***) zachodzi.

6. Zatem niezależnie od tego czy $NWD(m, p) = 1$, czy $NWD(m, p) > 1$, zawsze mamy

$$m^{ed} \equiv m(\mathrm{mod}\, p)\,.$$

7. Podobnie możemy rozumować dla liczby pierwszej q otrzymując

$$m^{ed} \equiv m(\mathrm{mod}\, q)$$

8. Korzystając z lematu (który jest prostym wnioskiem z chińskiego twierdzenia o resztach) dostajemy ostatecznie, że

$$m^{ed} \equiv m \ (\mathrm{mod}\, pq) \qquad (****)$$

co dowodzi tezy twierdzenia.

9. Z własności homomorfizmu (pierścieni Z i Z_n) jakim jest branie reszty z dzielenia przez n dostajemy jednocześnie z (****), że

$$m^{ed} = m \ (\mathrm{mod}\, pq)$$

gdzie podnoszenie do potęgi jest wielokrotnym mnożeniem modulo n w pierścieniu Z_n .■

Probabilistyczne uzasadnienie algorytmu RSA

Warto zwrócić uwagę na to, że nie wymagamy w założeniach powyższego twierdzenia by $NWD(m,n) = 1$ tak jak w założeniach twierdzenia Eulera. Próba uzasadniania kongruencji

$m^{ed} \equiv m \pmod{pq}$ jedynie twierdzeniem Eulera byłaby więc zręcznym korzystaniem z nieuwagi Czytelnika.

Jednak przy $n = p_1 \cdot p_2$ i założeniu równomiernego rozkładu prawdopodobieństwa na zbiorze Z_n prawdopodobieństwo tego, że $NWD(a,n) \neq 1$ maleje do 0 wraz ze wzrostem p_1 i p_2 . Istotnie z definicji funkcji Eulera mamy, że:

$$P(\{a \in Z_n; NWD(a,n) \neq 1\}) = \frac{n-\varphi(n)}{n} = \frac{n-(p_1-1)\cdot(p_2-1)}{n} = \frac{p_1+p_2-1}{p_1 \cdot p_2} = \frac{1}{p_1} + \frac{1}{p_2} - \frac{1}{n}$$

a więc $P(\{a \in Z_n; NWD(a,n) \neq 1\}) \to 0$ jeśli $p_1 \to +\infty$ i $p_2 \to +\infty$

zatem sytuacja, że $NWD(a,n) = 1$ nie zachodzi staje się bardzo mało prawdopodobna.

Jeśli jednak $NWD(m,n) = 1$ to rzeczywiście mamy bezpośrednio z twierdzenia Eulera

$$m^{(p-1)(q-1)} \equiv 1 \pmod{pq}$$

i dalej mnożąc obie strony tej kongruencji prze *m* dostajemy

$$m^{(p-1)(q-1)+1} \equiv m \pmod{pq}$$

co ponieważ $ed = 1 + k \cdot (p-1)(q-1)$ daje $m^{ed} \equiv m \pmod{pq}$

Jednak przy $n = p \cdot q$ (gdzie czynniki są różnymi liczbami pierwszymi) i założeniu równomiernego rozkładu prawdopodobieństwa na zbiorze Z_n prawdopodobieństwo tego, że $NWD(m,n) \neq 1$ maleje do 0 wraz ze wzrostem p i q. Istotnie z definicji funkcji Eulera mamy, że:

$$P(\{m \in Z_n;\ NWD(m,n) \neq 1\}) = \frac{n-\varphi(n)}{n} = \frac{n-(p-1)\cdot(q-1)}{n} = \frac{p+q-1}{p \cdot q} = \frac{1}{p} + \frac{1}{q} - \frac{1}{n}$$

a więc $P(\{m \in Z_n;\ NWD(m,n) \neq 1\}) \to 0$ jeśli $p \to +\infty$ i $q \to +\infty$.

Zatem sytuacja, że $NWD(m,n) = 1$ nie zachodzi staje się bardzo mało prawdopodobna.

Gdybyśmy więc nawet nie znali dowodu twierdzenia o poprawności RSA to możnaby powiedzieć, że dla dowolnego $\varepsilon > 0$ istnieje dostatecznie duże $N_0 \in N$ takie, że dla $p,q > N_0$ mamy

$$P(\{m \in Z_n;\ \ m^{ed} \equiv m \pmod{pq}\}) > 1 - \varepsilon$$

Powyższe rozumowanie stanowi „probabilistyczny dowód” twierdzenia o poprawności RSA. ■

Bezpieczeństwo szyfru RSA.

Bezpieczeństwo szyfru RSA jest zależne od tego czy istnieje efektywny algorytm rozkładu dużej liczby n na czynniki pierwsze czyli algorytm faktoryzacji. Dotychczas nie jest znany taki efektywny algorytm. Żeby złamać szyfr RSA wystarczy rozłożyć n na czynniki pierwsze. Znajomość rozkładu $n = pq$ umożliwia bowiem obliczenie wartości funkcji Eulera $\varphi(n) = (p-1)(q-1)$ co pozwala na proste obliczenie klucza prywatnego jako odwrotności klucza publicznego w pierścieniu $Z_{(p-1)(q-1)}$. Dobór odpowiednio dużych liczb pierwszych p i q praktycznie uniemożliwia faktoryzację n.

Podobnie w przypadku, gdy $n = p_1 p_2 \cdot \ldots \cdot p_k$ faktoryzacja n pozwala łatwo obliczyć wartość funkcji Eulera. mamy bowiem $\varphi(n) = (p_1 - 1)(p_2 - 2) \cdot \ldots \cdot (p_k - 1)$. Odtworzenie klucza prywatnego z klucza publicznego sprowadzałoby się w tej sytuacji do znalezienia elementu odwrotnego do klucza publicznego w pierścieniu $Z_{\varphi(n)}$ co sprowadza się z kolei do realizacji rozszerzonego algorytmu Euklidesa (istnieją szybkie efektywne wersje tego algorytmu np. binarny algorytm Euklidesa).

3. 3 System kryptograficzny Rabina

System kryptograficzny Rabina czyli szyfr Rabina jest klasycznym szyfrem z kluczem publicznym. Cechą wyróżniającą ten szyfr jest prostota algorytmu szyfrującego

Uwaga. Pożądaną własnością każdego systemu kryptograficznego jest wykazanie, że jego złamanie jest tak trudne jak rozwiązanie typowego problemu obliczeniowego powszechnie uważanego za trudny. Takimi typowymi dobrze zbadanymi trudnymi obliczeniowo problemami są np.: rozkład liczby całkowitej na czynniki pierwsze (faktoryzacja liczby całkowitej) lub problem logarytmów dyskretnych.

Jeśli chodzi o szyfr RSA, to wydaje się, że złamanie systemu RSA jest tak trudne obliczeniowo jak rozkład na czynniki pierwsze modułu n (względem którego podnosimy do potęgi). Ale nie jest to dowiedzione.

Szyfr Rabina jest przykładem systemu kryptograficznego, dla którego dowodzi się, że odtworzenie tekstu jawnego z szyfrogramu jest obliczeniowo równoważne rozkładowi liczby całkowitej na czynniki pierwsze.

Taki system kryptograficzny dla którego istnieje dowód, że złamanie tego systemu jest tak trudne jak rozwiązanie typowego problemu obliczeniowego powszechnie uważanego za trudny nosi nazwę systemu z dowodliwym bezpieczeństwem (ang. provably secure system). Zatem szyfr Rabina jest szyfrem z dowodliwym bezpieczeństwem.

Algorytm generacji klucza prywatnego i publicznego dla szyfru Rabina.

Strona A tworzy dla siebie parę kluczy (klucz publiczny, klucz prywatny) następująco:

1.Wybiera dwie różne liczby pierwsze p i q i oblicza $n = p \cdot q$. Liczby pierwsze p i q powinny być duże (o zbliżonej liczbie cyfr cyfr dziesiętnych np. około 100 dla utrudnienia faktoryzacji liczby n). Zakładamy, że problem faktoryzacji liczby n jest praktycznie nierozwiązywalny.

2.Przyjmuje, jako klucz prywatny (p,q) a jako klucz publiczny n
W zasadzie taki wybór klucza prywatnego i publicznego wystarczy. Żeby jednak uczynić procedurę deszyfracji bardziej efektywną zakłada się z reguły, że $p \equiv 3(\mathrm{mod}\,4)$ oraz $q \equiv 3(\mathrm{mod}\,4)$. Istnienie takich liczb pierwszych wynika z faktu, że w każdym (nietrywialnym) ciągu arytmetycznym o wyrazach naturalnych mamy nieskończenie wiele liczb pierwszych.

Algorytm szyfrowania dla szyfru Rabina.

Strona B szyfruje wiadomość jawną $m \in Z_n$ dla strony A następująco:

1.Wstępnie strona B musi uzyskać klucz publiczny strony A czyli n a następnie wiadomość jawną $m \in Z_n$ reprezentuje jako liczbę z pierścienia Z_n.

2.Teraz strona *B* oblicza szyfrogram jako:

$$c = m^2 \pmod n$$

i wysyła szyfrogram *c* do *A*. Algorytm szyfrowania jest więc wyjątkowo prosty.

Algorytm deszyfrowania dla szyfru Rabina.

Strona *A* deszyfruje szyfrogram $c \in Z_n$ odzyskując wiadomość jawną $m \in Z_n$ następująco:

1.obliczamy cztery pierwiastki kwadratowe m_1, m_2, m_3, m_4 z *c* modulo *n*. Istnieją dokładnie cztery, jeśli $NWD(m,n) = 1$. Przypadek $NWD(m,n) > 1$ jest jak łatwo sprawdzić dla dużych *n* mało prawdopodobny (przy równomiernym prawdopodobieństwie wyboru liczby *m* ze zbioru Z_n), ale możliwy. Mamy wówczas 2 pierwiastki lub 1 pierwiastek.

2.Wiadomość jawna *m* jest na pewno jedną z wiadomości m_1, m_2, m_3, m_4. Nadając pewne cechy charakterystyczne szyfrowanej wiadomości jawnej *m*, możemy na ogół bez trudu wybrać spośród m_1, m_2, m_3, m_4 właściwe *m*.

Konieczność wyboru właściwego *m* spośród kilku możliwości jest pewną wadą szyfru Rabina.

Algorytm obliczania pierwiastków kwadratowych z $c \in Z_n$ modulo $n = p \cdot q$, gdzie *p* i *q* to dwie różne liczby pierwsze takie, że $p \equiv 3 \pmod 4$, $q \equiv 3 \pmod 4$

1. Znaleźć $a, b \in Z$ takie, że $a \cdot p + b \cdot q = 1$ (np. za pomocą rozszerzonego algorytmu Euklidesa). Warto zauważyć, że wystarczy takie liczby a, b znaleźć tylko raz. Są one w gruncie rzeczy parametrami tworzonego systemu kryptograficznego.
2. Obliczyć $r = c^{(p+1)/4} \pmod p$.
3. Obliczyć $s = c^{(q+1)/4} \pmod q$.
4. Obliczyć $x = (aps + bqr) \pmod n$.
5. Obliczyć $y = (aps - bqr) \pmod n$
6. Cztery pierwiastki kwadratowe z *c* modulo *n* są takie:
   $x, -x \pmod n$, $y, -y \pmod n$.

Dowód. Dowód polega na zwykłym sprawdzeniu czy podniesienie do kwadratu da *c* tzn. czy $x^2 \pmod n = ((aps + bqr))^2 \pmod n = c$ oraz czy $y^2 \pmod n = ((aps - bqr))^2 \pmod n = c$ ■

Bezpieczeństwo algorytmu Rabina

Odzyskanie wiadomości jawnej $m \in Z_n$, gdy mamy tylko kryptogram $c = m^2 (\mathrm{mod}\, n)$ (czyli sytuacja podsłuchu pasywnego) jest dokładnie problemem obliczania pierwiastka kwadratowego z m w Z_n. Bezpieczeństwo szyfru Rabina wynika z faktu, że obliczenie pierwiastka kwadratowego w Z_n, czyli znalezienie dla danej liczby $a \in Z_n$ takiej liczby $x \in Z_n$, że

$$x^2 \equiv a (\mathrm{mod}\, n)$$

(podnosimy x do kwadratu w zbiorze liczb całkowitych Z) lub równoważnie

$$x^2 = a \quad \text{w} \quad Z_n$$

(podnosimy x do kwadratu w Z_n) jest ogólnie rzecz biorąc obliczeniowo trudne.

Zachodzi bowiem następujący fakt:

Fakt. Problem faktoryzacji liczby naturalnej n i obliczanie pierwiastków kwadratowych w Z_n są ze sobą równoważne.

Zatem bezpieczeństwo szyfru Rabina wynika z faktu, że problem faktoryzacji liczby naturalnej jest obliczeniowo trudny.

3. 4 System kryptograficzny ElGamala

System kryptograficzny ElGamala jest typowym systemem kryptograficznym z kluczem publicznym.

Założenia. Ustalamy bardzo duże ciało skończone F_q (a dokładniej $(F_q,\oplus,\otimes)$, gdzie $\oplus$ jest dodawaniem w ciele a $\otimes$ mnożeniem w ciele) i wybieramy jeden jego element $g \in F_q^*$ z grupy multiplikatywnej F_q^* (najlepiej, żeby to był generator grupy multiplikatywnej F_q^* ale nie jest to konieczne). Wiadomości jawne to elementy ciała F_q.

Algorytm generacji klucza prywatnego i publicznego dla szyfru ElGamala.

Użytkownik A wybiera losowo liczbę całkowitą $a \in <2, q-2>$ (ta liczba będzie tajnym kluczem rozszyfrowującym tworzone kryptogramy) a następnie podnosi g do potęgi a w ciele F_q i ujawnia $g^a \in F_q$ (g^a będzie służyło do szyfrowania czyli do tworzenia kryptogramów). Kluczem prywatnym jest wybrana liczba $a \in <2, q-2>$ a kluczem publicznym jest g^a. Reasumując, ogólnie znane są trzy liczby: liczby g, g^a oraz liczba q natomiast liczba $a \in <2, q-2>$ pozostaje tajna.

Algorytm szyfrowania dla szyfru ElGamala.

Załóżmy, że nadawca B chce wysłać wiadomość m (jest to liczba z ciała F_q) do odbiorcy A. Wybiera on losowo liczbę $k \in <2, q-2>$ i przesyła do A parę uporządkowaną liczb

$$(g^k, m \otimes ((g^a))^k)$$

Podnoszenie do potęgi odbywa się w ciele F_q. Warto również przypomnieć w tym miejscu, że dla podnoszenia do potęgi w ciele istnieją szybkie, efektywne algorytmy.

Losowy wybór liczby $k \in <2, q-2>$ powoduje, że kryptogram dla ustalonej wiadomości jawnej m jest realizacją pewnej zmiennej losowej co oczywiście utrudnia atak kryptoanalitykowi.

Algorytm deszyfrowania dla szyfru ElGamala.

1. Odbiorca A podnosi pierwszą liczbę do potęgi a (liczba a jest kluczem prywatnym strony A) otrzymując liczbę g^{ak} a następnie wykorzystując np. rozszerzony algorytm Euklidesa oblicza g^{-ak} czyli odwrotność liczby g^{ak} w ciele F_q.

2. Odbiorca A mnoży drugą z przesłanych liczb czyli $m \otimes_q ((g^a))^k$ przez g^{-ak} uzyskując w wyniku tego mnożenia liczbę m stanowiącą tekst jawny. Istotnie z łączności mnożenia w ciele F_q wynika, że:

$$(m \otimes g^{ak}) \otimes g^{-ak} = m \otimes (g^{ak} \otimes g^{-ak}) = m \otimes 1 = m$$

Bezpieczeństwo szyfru ElGamala.

Zauważmy, że gdyby istniał efektywny algorytm rozwiązujący problem logarytmu dyskretnego to znając klucz publiczny można by znaleźć łatwo klucz prywatny a.

Bezpieczeństwo szyfru ElGamala wynika więc z faktu, że problem logarytmu dyskretnego jest trudny obliczeniowo.

Bardzo duże ciało skończone F_q, w którym prowadzimy obliczenia wybierane jest tak by problem logarytmu dyskretnego był w nim praktycznie nierozwiązywalny.

Podsumowanie

1. Obliczenia przeprowadzamy w ciele F_q. Ogólnie znane są 3 liczby $g, g^a \in F_q^*$ oraz liczba elementów ciała q, natomiast tajne jest a i to jest klucz prywatny rozszyfrowujący kryptogram.

2.Przesyłamy odbiorcy wiadomości parę uporządkowaną liczb $(g^k, P \otimes_q ((g^a))^k)$.

3.5. Szyfry z kluczem publicznym oparte na problemie plecakowym czyli szyfry plecakowe

1. Problem plecakowy (inaczej problem upakowania lub problem sumy) ang. knapsack problem jest następujący. Dany jest ciąg skończony liczb naturalnych $(a_1, a_2, ..., a_n)$, który nazywamy wektorem plecakowym (lub plecakiem) oraz liczba naturalna k. Poszukujemy takich liczb $i_1, i_2, ..., i_n \in \{0,1\}$, żeby :

$$\sum_{j=1}^{n} i_j a_j = k . \qquad (1)$$

Liczby $a_1, a_2, ..., a_n$ nie muszą być parami różne.

Wiadomo że:

Twierdzenie . Problem plecakowy jest *NP*-zupełny.

Pochodzenie nazwy problemu jest oczywiste. Niech będzie dany zbiór n przedmiotów każdy o pewnej wadze $a_1, a_2, ..., a_n$, chcemy niektóre z nich (być może wszystkie) zapakować do plecaka tak by plecak ważył dokładnie k. Naturalne jest pytanie czy jest to możliwe a jeśli tak to jakie przedmioty wybrać. Czas wymagany do rozwiązania tego problemu rośnie bardzo szybko wraz ze wzrostem n. Najprostszy algorytm rozwiązujący problem plecakowy to algorytm przeglądania wszystkich możliwych ciągów binarnych $(i_1, i_2, ..., i_n)$ i sprawdzania czy równość (1) zachodzi ale wszystkich takich ciągów mamy 2^n.

Przykład. Przedmioty te mogą ważyć np. 1, 5, 6, 11, 14 i 20 (jednostek wagi). Jest możliwe zapakowanie plecaka tak, aby ważył 22 należy wziąć przedmioty o wadze 5, 6 i 11. Istotnie 5+6+11=22. Nie jest możliwe spakowanie plecaka tak, aby ważył 24. By to wykazać wystarczy sprawdzić, że dla każdego słowa binarnego $(i_1, i_2, ..., i_n) \in \{0,1\}^6$ mamy $i_1 + 5i_2 + 6i_3 + 11i_4 + 14i_5 + 20i_6 \neq 24$. Zatem problem plecakowy może mieć rozwiązanie a może go nie mieć zależnie od danych precyzujących problem czyli od $(a_1, a_2, ..., a_n)$ i k. Ogólnie rzecz biorąc dany problem plecakowy może mieć dokładnie jedno rozwiązanie może mieć wiele rozwiązań lub nie mieć ich wcale.

2. Pierwszy system kryptograficzny z kluczem publicznym oparty na problemie plecakowym opracowali w 1978 r. Ralph Merkle i Martin Hellman stąd nazwa tego systemu: szyfr Merklego-Hellmana. Mógł on być używany tylko do szyfrowania, chociaż Shamir adaptował później ten system do podpisu cyfrowego. Systemy kryptograficzne z kluczem publicznym oparte na problemie plecakowym nazywa się też krócej szyframi lub algorytmami plecakowymi (ang. knapsack algorithms). Bezpieczeństwo algorytmów plecakowych wynika z faktu, że problem plecakowy jest trudny obliczeniowo.

Uwaga. Istnieje wiele metod szyfrowania opartych na problemie plecakowym. Większość z nich została jednak złamana. Omówimy dwie metody tzw. szyfr Merklego-Hellmana oraz szyfr Chora-Rivesta.

3. Szyfr Merklego-Hellmana (lub algorytm plecakowy Merklego-Hellmana). Chociaż stwierdzono, że klasyczny szyfr Merklego-Hellmana nie jest bezpieczny (metody łamania tego szyfru zostały podane przez A.Shamira) warto go przedstawić, ponieważ pokazuje, w jaki sposób problem NP - zupełny może być wykorzystany w kryptografii z kluczem publicznym

Szyfrowanie za pomocą wektorów plecakowych: ciąg bitów $i_1, i_2, ..., i_n \in \{0,1\}$ jest kodowany przez liczbę $\sum_{j=1}^{n} i_j a_j = k$ powstają przy tym jednak następujące problemy:

- Pojedynczy kryptogram musi odpowiadać jednemu tekstowi jawnemu. Innymi słowy chcemy by funkcja szyfrująca $f : \{0,1\}^n \ni (i_1, i_2, ..., i_n) \to f(i_1, i_2, ..., i_n) = \sum_{j=1}^{n} i_j a_j \in N$ była różnowartościowa. Jeśli nie nałożymy specjalnych ograniczeń na wektor plecakowy $(a_1, a_2, ..., a_n)$ to może się jednak zdarzyć, że dwa różne słowa binarne $(i_1, i_2, ..., i_n)$ i $(i_1', i_2', ..., i_n')$ mają ten sam kryptogram czyli $\sum_{j=1}^{n} i_j a_j = \sum_{j=1}^{n} i_j' a_j$.

- Co prawda złamanie takich szyfrów jest trudne ze względu na NP zupełność problemu plecakowego, ale nie znamy również żadnej szybkiej metody, by za pomocą klucza taki kryptogram deszyfrować.

Oba problemy dają się rozwiązać poprzez ograniczenie nałożone na wektory plecakowe.

Wektory plecakowe superrosnące: wektor plecakowy $(a_1, a_2, ..., a_n)$ gdzie $n \geq 2$ nazywamy superrosnącym, jeśli dla każdego $i = 2,3,...,n$ mamy:

$$\sum_{j<i} a_j < a_i \qquad (*)$$

Wektory plecakowe superrosnące nazywamy krócej wektorami superrosnącymi lub plecakami superrosnącymi. Analogicznie możemy mówić, że ciąg o wartościach w N jest superrosnący jeśli dla każdego $i = 2,3,...,n,...$ spełniony jest warunek (*). Warunek (*) oznacza, że każdy element ciągu (oprócz pierwszego) jest większy od sumy wszystkich poprzedzających go elementów.

Przykład. Ciąg skończony 1, 3, 6, 13, 27, 52 jest ciągiem superrosnącym, natomiast ciąg 1, 3, 4, 9, 15, 25 nim nie jest. Łatwo również sprawdzić, że ciągami superrosnącymi są ciągi $(2^{n-1})_{n=1}^{\infty}$, $(10^{n-1})_{n=1}^{\infty}$, czy ogólniej $(W^{n-1})_{n=1}^{\infty}$, gdzie $W \geq 2, W \in N$.

Lemat. Jeśli $(a_1, a_2, ..., a_n)$ jest wektorem plecakowym superrosnącym to

1. każdemu kryptogramowi odpowiada dokładnie jeden tekst jawny, czyli funkcja szyfrująca

$$f : \{0,1\}^n \ni (i_1, i_2, ..., i_n) \to f(i_1, i_2, ..., i_n) = \sum_{j=1}^{n} i_j a_j \in N$$

jest różnowartościowa

2. deszyfrowanie może być dokonane w czasie proporcjonalnym do długości *n* wektora superrosnącego.

Dowód. Pierwsza część lematu wynika z drugiej. Niech $(a_1, a_2, ..., a_n)$ będzie danym wektorem superrosnącym a liczba naturalna $k = \sum_{j=1}^{n} i_j a_j$ niech będzie kryptogramem dla nieznanego słowa binarnego $(i_1, i_2, ..., i_n)$. Pokażemy, że mając k i $(a_1, a_2, ..., a_n)$ łatwo odtworzymy $(i_1, i_2, ..., i_n)$.

Zauważmy najpierw , że

$i_n = 1$ wtedy i tylko wtedy, gdy $k \geq a_n$ (bowiem $\sum_{j<n} a_j < a_n$).

Zatem i_n daje się łatwo wyliczyć z k (wystarczy jedno porównanie k z a_n). Po znalezieniu i_n rozważamy $k' = k - i_n \cdot a_n$ i superrosnący wektor $(a_1, a_2, ..., a_{n-1})$. Kryptogram *k'* odpowiada temu wektorowi i tekstowi jawnemu $(i_1, i_2, ..., i_{n-1})$. Analogicznie jak poprzednio możemy znaleźć teraz i_{n-1}. Postępowanie to kontynuujemy, aż ustalone zostaną jednoznacznie wszystkie bity i_j. Zatem funkcja szyfrująca $f : \{0,1\}^n \to N$ jest różnowartościowa. Oczywiście zaproponowany wyżej algorytm odtwarzania słowa binarnego $(i_1, i_2, ..., i_n)$ ma liniową względem *n* złożoność obliczeniową. Deszyfrowanie może być więc dokonane w czasie proporcjonalnym do długości wektora superrosnącego ■

W powyższym dowodzie zaproponowany został następujący algorytm deszyfracji dla wektora plecakowego superrosnącego

Algorytm deszyfracji dla wektora superosnącego $(a_1, a_2, ..., a_n)$

DANE WEJŚCIOWE: kryptogram $k \in N \cup \{0\}$, $n \in N$ wektor superrosnący $(a_1, a_2, ..., a_n)$
DANE WYJŚCIOWE: słowo binarne $(i_1, i_2, ..., i_n)$ będące tekstem jawnym

for $j := n$ **downto** 1 **begin if** $k \geq a_j$ **then** $i_j := 1$ **else** $i_j := 0$;
$k := k - a_j$
end

Problem. Dzięki wektorom superrosnącym kryptogramy odpowiadają jednoznacznie tekstom jawnym. Z drugiej jednak strony, każdy kto zna taki wektor, $(a_1, a_2, ..., a_n)$ może zarówno łatwo szyfrować, jak i łatwo deszyfrować. Otrzymujemy więc dla wektora superrosnącego system kryptograficzny z kluczem prywatnym (dokładniej z kluczem symetrycznym). Kluczem jest tu oczywiście wektor superrosnący $(a_1, a_2, ..., a_n)$. Nie są zatem spełnione wymagania nakładane na asymetryczne algorytmy z kluczem publicznym.

Utajnianie wektora superrosnącego $(a_1, a_2, ..., a_n)$: Aby poradzić sobie z problemem opisanym powyżej, szyfrowanie będzie się odbywać za pomocą innego wektora plecakowego niesuperrosnącego $(a_1', a_2', ..., a_n')$, który uzyskamy z wektora superosnącego $(a_1, a_2, ..., a_n)$. W tym celu wybieramy dwie liczby naturalne M i W spełniające następujące dwa warunki:

1) $M > \sum_{i \leq n} a_i$

2) $W < M$, takie że $W > 1$ oraz NWD(W, M) = 1

Niech $a_i' = a_i \cdot W \pmod{M}$. Zauważmy, że ciąg $(a_1', a_2', ..., a_n')$ nie musi być superrosnący. choć może się tak zdarzyć. Chcemy tak dobrać liczbę W, $1 < W < M$ by ciąg $(a_1', a_2', ..., a_n')$ nie był superrosnący. Odpowiednia liczba W musi być na tyle duża, żeby $a_n \cdot W > M$, jest to oczywisty warunek konieczny tego by ciąg $(a_1', a_2', ..., a_n')$ nie był superosnący.

Czasami sugerowany jest w literaturze jako dobry wybór W wybór takiego W, które spełnia warunek $W > M/2$, ale jak łatwo sprawdzić warunek ten nie jest warunkiem wystarczającym do tego, żeby ciąg $(a_1', a_2', ..., a_n')$ nie był superosnący.

Załóżmy, że udało się nam dobrać takie W, że ciąg $(a_1', a_2', ..., a_n')$ nie jest superosnący. Ciąg ten następnie permutuje się permutacją $\pi : \{1,2,...,n\} \to \{1,2,...,n\}$ i tak otrzymany wynik podaje jako k1ucz publiczny $(a_{\pi(1)}', a_{\pi(2)}', ..., a_{\pi(n)}')$. Kluczem prywatnym jest wektor superrosnący $(a_1, a_2, ..., a_n)$ liczby M oraz W, a ponadto użyta permutacja $\pi : \{1,2,...,n\} \to \{1,2,...,n\}$ czyli czwórka uporządkowana $((a_1, a_2, ..., a_n), M, W, \pi)$. Liczby W i M są względnie pierwsze, więc za pomocą rozszerzonego algorytmu Euklidesa lub z twierdzenia Eulera możemy wyznaczyć liczbę W^{-1} w pierścieniu $Z_M = \{0,1,...,M-1\}$.

Ponieważ przy deszyfracji nie używamy liczby W a W^{-1} nieco wygodniej uważać za klucz prywatny czwórkę uporządkowaną: $((a_1,a_2,...,a_n),M,W^{-1},\pi)$.

Szyfrowanie i deszyfrowanie:

Szyfrowanie słowa binarnego $(i_1,i_2,...,i_n)$ odbywa się standardowo za pomocą ciągu będącego kluczem publicznym $(a'_{\pi(1)},a'_{\pi(2)},...,a'_{\pi(n)})$: kryptogram jest równy liczbie $\sum_{j=1}^{n} i_j a'_{\pi(j)}$.

Deszyfrowanie. Niech $y = i_1 \cdot a'_{\pi(1)} + i_2 \cdot a'_{\pi(2)} + ... + i_n \cdot a'_{\pi(n)}$ będzie rozważanym kryptogramem, dla którego szukamy $i_1,i_2,...,i_n$. Wykonujemy następujące dwa kroki:

1. Obliczamy $W^{-1} \cdot y \pmod{M}$ czyli $W^{-1} \otimes_M [y]_M$

2. Znając liczbę $W^{-1} \otimes_M [y]_M$ i wektora superrosnącego $(a_1,a_2,...,a_n)$ znajdujemy łatwo tekst jawny $i_1,i_2,...,i_n$ stosując opisany już algorytm deszyfracji dla wektora superrosnącego. Jak zauważyliśmy poprzednio, można zrealizować ten algorytm w czasie proporcjonalnym do długości wektora superrosnącego tzn. w czasie proporcjonalnym do n.

Poniższe twierdzenie jest twierdzeniem o poprawności deszyfracji.

Twierdzenie. Niech $(a_1,a_2,...,a_n)$ będzie wektorem plecakowym superrosnącym, M dowolną liczbą naturalną taką, że $M > \sum_{i \le n} a_i$, $\pi : \{1,2,...,n\} \to \{1,2,...,n\}$ dowolną permutacją zbioru $\{1,2,...,n\}$ a W liczbą naturalną taką, że $W \in Z_M = \{0,1,...,M-1\}$, $W \ge 2$ i $NWD(W,M) = 1$. Niech ponadto dla każdego $i \in \{1,2,...,n\}$ mamy $a'_i = a_i \cdot W \pmod{M}$ (czy równoważnie dla każdego $i \in \{1,2,...,n\}$ $a'_i = a_i \otimes_M W$) wówczas:

1) Funkcja szyfrująca f dla wektora plecakowego $(a'_{\pi(1)},a'_{\pi(2)},...,a'_{\pi(n)})$ zadana wzorem:
$f : \{0,1\}^n \ni (i_1,i_2,...,i_n) \to f(i_1,i_2,...,i_n) = \sum_{j=1}^{n} i_j a'_{\pi(j)} \in N$ jest różnowartościowa

2) Mając klucz prywatny $((a_1,a_2,...,a_n),M,W^{-1},\pi)$ możemy dokonać deszyfracji kryptogramu w czasie proporcjonalnym do n.

Dowód. Obliczamy $W^{-1} \otimes_M [y]_M$ (łatwo widać, że $y < M$ i $y \in Z_M$). Liczba ta jest równa

$$W^{-1} \otimes_M [y]_M = W^{-1} \otimes_M (i_1 \otimes_M a'_{\pi(1)} \oplus i_2 \otimes_M a'_{\pi(2)} \oplus_M \ldots \oplus_M i_n \otimes_M a'_{\pi(n)}) =$$
$$= W^{-1} \otimes_M (i_1 \otimes_M a_{\pi(1)} \otimes_M W \oplus_M i_2 \otimes_M a_{\pi(2)} \otimes_M W \oplus_M \ldots \oplus_M i_n \otimes_M a_{\pi(n)} \otimes_M W) =$$
$$= i_1 \cdot a_{\pi(1)} + i_2 \cdot a_{\pi(2)} + \ldots + i_n \cdot a_{\pi(n)} = i_{\pi^{-1}(1)} \cdot a_1 + i_{\pi^{-1}(2)} \cdot a_2 + \ldots + i_{\pi^{-1}(n)} \cdot a_n$$

Stosując teraz algorytm deszyfracji dla wektora superrosnącego $(a_1, a_2, \ldots, a_n)$ (por. lemat) odtwarzamy jednoznacznie słowo binarne $(i_{\pi^{-1}(1)}, i_{\pi^{-1}(2)}, \ldots, i_{\pi^{-1}(n)}) \in \{0,1\}^n$. Permutując permutacją $\pi : \{1,2,\ldots,n\} \to \{1,2,\ldots,n\}$ wyrazy tak uzyskanego ciągu wyznaczamy jednoznacznie tekst jawny $(i_1, i_2, \ldots, i_n) \in \{0,1\}$. Wynika stąd różnowartościowość funkcji

$$f : \{0,1\}^n \ni (i_1, i_2, \ldots, i_n) \to f(i_1, i_2, \ldots, i_n) = \sum_{j=1}^{n} i_j a'_{\pi(j)} \in N .$$

Liniowa złożoność obliczeniowa algorytmu deszyfrującego wynika z p.2 lematu. ■

Jak już wspominaliśmy, przedstawiony algorytm nie powinien być stosowany w praktyce, gdyż została znaleziona szybka metoda deszyfrowania kryptogramu bez znajomości tajnego klucza. Nie oznacza to, że znaleziony został szybki algorytm rozwiązujący NP-zupełny problem plecakowy. Algorytm łamiący szyfr plecakowy pracuje tylko dla specjalnych wektorów $(a'_1, a'_2, \ldots, a'_n)$ uzyskanych z wektorów superrosnących $(a_1, a_2, \ldots, a_n)$. Dla dowolnych wektorów $(a_1, a_2, \ldots, a_n)$ dalej nie jest znana żadna efektywna metoda rozwiązania problemu plecakowego.

Algorytm plecakowy Chora-Rivesta.

Warto wspomnieć, że istnieje odmiana algorytmu plecakowego, która jest obecnie bezpieczna (tzn. algorytm nie został dotychczas złamany) jest to tzw. algorytm plecakowy Chora-Rivesta. Ilość obliczeń wymaganych do realizacji tego algorytmu jest jednak stosunkowo duża.

3.6 System kryptograficzny Massey'a –Omury

System kryptograficzny Masseya-Omury służy do szyfrowania przesyłanej informacji. Jest to system z kluczem asymetrycznym.

1. Wybieramy ciało skończone F_q wspólne dla *r* użytkowników systemu i każdy użytkownik losuje ze zbioru $Z_{q-1} = \{0,1,...,q-1\}$ liczbę e_i $(i = 1,2,...,r)$ taką, by $NWD(e_i, q-1) = 1$, jest to liczba tajna, (tzn. e_i znana jest tylko *i* – temu użytkownikowi).

2. Każdy użytkownik oblicza sobie liczbę d_i odwrotną do e_i w pierścieniu Z_{q-1}, czyli inaczej takie d_i, że $d_i \cdot e_i \equiv 1(\mathrm{mod}(q-1))$ lub $d_i \otimes_{q-1} e_i = 1$.

3. Przesłanie informacji np. od użytkownika A_1 do A_2 odbywa się tak:

I. A_1 podnosi wiadomość jawną m do potęgi e_1 tzn. oblicza m^{e_1} w ciele F_q i przesyła do A_2.

II A_2 podnosi m^{e_1} do potęgi e_2 tzn. oblicza $(m^{e_1})^{e_2}$ w ciele F_q i przesyła do A_1.

III A_1 podnosi $(m^{e_1})^{e_2}$ do potęgi d_1 tzn. oblicza $((m^{e_1})^{e_2})^{d_1} = (m)^{d_1}$ w ciele F_q i przesyła do A_2.

IV A_2 podnosi $m^{e_1 e_2 d_1}$ do potęgi d_2 tzn. oblicza $(((m^{e_1})^{e_2})^{d_1})^{d_2}$ i znajduje *m* ponieważ $(((m^{e_1})^{e_2})^{d_1})^{d_2} = m$.

Uzasadnienie poprawności. Korzystając z twierdzenia, że rząd dowolnego elementu $a \in F_q^*$ dzieli rząd grupy multiplikatywnej ciała (czyli $q-1$) dostajemy

$$(((m^{e_1})^{e_2})^{d_1})^{d_2} = (m^{e_1 d_1})^{e_2 d_2} = (m^{1+k_1\cdot(q-1)})^{1+k_2\cdot(q-1)} = m$$

Uwaga 1. Ten system wymaga dobrej metody sprawdzania tożsamości strony A_2, (osoby odczytującej informacje) bo może się ktoś pod nią bardzo łatwo podszyć. Bierny podsłuch jest natomiast nieskuteczny. W praktyce konieczne jest więc uzupełnienie systemu Masseya-o jakieś mechanizmy uwierzytelniania np. podpisy cyfrowe.

Bezpieczeństwo systemu Masseya-Omury. Bezpieczeństwo systemu wynika ze złożoności obliczeniowej algorytmu obliczenia logarytmu dyskretnego. Zauważmy, że strona A_2 poznaje wiadomość jawną m i m^{e_1} zatem jeśli potrafi obliczyć $\log_m m^{e_1}$ to zna e_1 czyli tajny klucz prywatny strony A_1.

Literatura

[1] A. Menezes, P. Oorschot, S. Vanstone; Handbook of Applled Cryptography; CRC Press Inc., 1997. (treść jest na stronie: http://cacr.math.uwaterloo.ca/hac)

[2] J.Stokłosa, T.Bilski,T.Pankowski; Bezpieczeństwo danych w systemach informatycznych; PWN, Warszawa 2001.

[3] N.Koblitz; Algebraiczne aspekty kryptografii; WNT, Warszawa 2000.

[4] N.Koblitz; A Course in Number Theory and Cryptography; Springer Verlag, New York 1994. (jest przekład polski p.t. Wykład z teorii liczb i kryptografii; WNT, Warszawa 1995.)

[5] M.Kutyłowski, W.Strothmann; Kryptografia, teoria i praktyka zabezpieczania systemów komputerowych; Oficyna Wydawnicza Read Me, Warszawa 2001.

[6] B.Schneier; Kryptografia dla praktyków; WNT, 2002.

[7] T.HCormen, C.E.Leiserson, R.L.Rivest; Wprowadzenie do algorytmów; WNT, 2004.

Zadania

Zadanie 1

Niech $n = p_1 \cdot p_2$, gdzie p_1, p_2 są liczbami pierwszymi. Niech ponadto $(Z_n, 2^{Z_n}, P)$ będzie przestrzenią probabilistyczną taką, że rozkład prawdopodobieństwa P jest równomierny tzn.

$$P(\{a\}) = \frac{1}{n} \text{ dla każdego } a \in Z_n .$$

Pokazać, że jeśli $p_1, p_2 \to +\infty$ to:

$$P(\{a \in Z_n ; NWD(a,n) > 1) \to 0$$

Uwaga. Jeśli $NWD(a,n) = 1$ oraz $e, d \in Z_{\varphi(n)}$ są względem siebie odwrotne w $Z_{\varphi(n)}$ (gdzie φ jest funkcją Eulera), to dla dowodu, że

$$a^{e \cdot d} = a \qquad (*)$$

wystarczy skorzystać (por. zadanie xx) z twierdzenia Eulera: jeśli $NWD(a,n) = 1$, to $a^{\varphi(n)} \equiv 1 \pmod{n}$.

Gdybyśmy równości (*) nie potrafili wykazać dla każdego $a \in Z_n$ (na szczęście potrafimy to robić, por. zadanie xx), to sytuacja wątpliwa (tzn. gdy $NWD(a,n) > 1$ i nie możemy tak jak wyżej zastosować twierdzenia Eulera) zachodziłaby dla dostatecznie dużych p_1, p_2 z bardzo małym prawdopodobieństwem. Stanowi to „probabilistyczny dowód poprawności szyfru *RSA*".

Rozwiązanie

1. Jeśli p_1, p_2 są różnymi liczbami pierwszymi to ponieważ $\varphi(n) = (p_1 - 1)(p_2 - 1)$ mamy dalej:

$$P(\{a \in Z_n ; NWD(a,n) > 1\}) = P(Z_n \setminus \{a \in Z_n ;\ NWD(a,n) = 1\}) =$$

$$1 - P(\{a \in Z_n ; NWD(a,n) = 1\}) = 1 - \frac{\varphi(n)}{n} = \frac{n - (p_1 - 1)(p_2 - 1)}{n} =$$

$$= \frac{n - p_1 \cdot p_2 - 1 + p_1 + p_2}{n} = \frac{1}{p_2} + \frac{1}{p_2} - \frac{1}{p_1 \cdot p_2}$$

Zatem jeśli $p_1, p_2 \to +\infty$, to $P(\{a \in Z_n ; NWD(a,n) > 1\}) \to 0$.

2. Jeśli $p_1 = p_2$ to $\varphi(n) = (p_1 - 1)p_1$ i analogicznie jak poprzednio dostajemy: jeśli $p_1, p_2 \to +\infty$, to $P(\{a \in Z_n; NWD(a,n) > 1\}) \to 0$. ■

Zadanie 2

Pokazać, że jeśli $n = p_1 \cdot p_2 \cdot ... \cdot p_r$, gdzie p_1 dla $i = 1,2,...,r$ są różnymi liczbami pierwszymi (mówimy w tej sytuacji, ze liczba *n* jest bezkwadratowa) oraz liczba $a \in Z_n$ i $NWD(a,n) = 1$ a ponadto $e,d \in Z_{\varphi(n)}$ są względem siebie odwrotne w pierścieniu $Z_{\varphi(n)}$ (gdzie φ jak zwykle jest funkcją Eulera), to

$$a^{ed} = a$$

(podnosimy do potęgi w pierścieniu Z_n).

Rozwiązanie

Istnieją takie $k \in Z$, że

$$a^{ed} = e^{[ed]_{\varphi(n)} + k \cdot \varphi(n)}$$

Z własności homomorfizmu jakim jest branie reszty modulo mamy:

$$[e \cdot d]_{\varphi(n)} = e \otimes_{\varphi(n)} d$$

i dalej z twierdzenia Eulera mamy

$$a^{ed} = e^{[ed]_{\varphi(n)} + k \cdot \varphi(n)} = a^{e \otimes_{\varphi(n)} d} \cdot (a^{\varphi(n)})^k = a^{e \otimes_{\varphi(n)} d} = a^1 = a$$

Ostatecznie więc $a^{ed} = a$.■

Zadanie 8.3
Niech $(Z_{\varphi(n)}, 2^{Z_{\varphi(n)}}, P)$ będzie przestrzenią probabilistyczną z równomiernym rozkładem P. Symbol φ oznacza tu funkcję Eulera. Niech ponadto $n = p_1 p_2 ... p_r$ gdzie p_i dla $i = 1,2,...,r$ są różnymi liczbami pierwszymi (n jest więc liczbą bezkwadratową). Znaleźć prawdopodobieństwo

$$P(\{a \in Z_{\varphi(n)}; a \text{ ma element odwrotny w } Z_{\varphi(n)}\})$$

czyli mówiąc niezbyt precyzyjnie, chodzi o znalezienie prawdopodobieństwa tego, że losowo wybrany z $Z_{\varphi(n)}$ element ma element odwrotny w $Z_{\varphi(n)}$.

Uwaga. Sytuację taką jak w zadaniu mamy przy losowaniu klucza prywatnego dla szyfru RSA.

Rozwiązanie:
Ogólnie rzecz biorąc element a pierścienia Z_m ma element odwrotny wtedy i tylko wtedy gdy $NWD(a,m) = 1$ zatem

$P(\{a \in Z_{\varphi(n)}; a \text{ ma element odwrotny w } Z_{\varphi(n)}\}) =$

$$= P(\{a \in Z_{\varphi(n)}; NWD(a, \varphi(n)) = 1\}) = \frac{\varphi(\varphi(n))}{\varphi(n)} = \qquad (*)$$
$$= \frac{\varphi((p_1 - 1)(p_2 - 1) \cdot ... \cdot (p_r - 1))}{(p_1 - 1)(p_2 - 1) \cdot ... \cdot (p_r - 1)}$$

Uwaga. Wzór (*) podaje prawdopodobieństwo tego, że nie będziemy musieli dokonywać powtórnego losowania klucza przy ustalaniu parametrów szyfru RSA. ■

Zadanie 4
Załóżmy że liczba $n \in N$ jest bezkwadratowa tzn. $n = p_1 p_2 p_3 ... p_k$, gdzie $p_1, p_2, p_3, ..., p_k$ są różnymi liczbami pierwszymi i k>1. Niech dwie liczby naturalne $d, e \in N$ będą takie, że $de - 1$ jest podzielne przez $p_i - 1$ dla każdego $i \in <1, k>$ wykazać, że wówczas dla każdego $a \in Z$ mamy

$$a^{de} \equiv a \pmod{n} \qquad (*)$$

lub równoważnie, dla każdego $a \in Z_n$ mamy $a^{de} = a$ w pierścieniu Z_n (potęgujemy w pierścieniu Z_n). Powyższy fakt jest podstawowy dla dowodu poprawności szyfru RSA.

Rozwiązanie

1. Jeśli a=0 to oczywiście $a^{de} \equiv a \pmod{n}$. Pozostaje sprawdzenie kongruencji (*) dla $a \in Z, a \neq 0$. Niech więc $a \in Z, a \neq 0$. Bezpośrednio z założenia mamy, że dla każdego $i \in <1, k>$ istnieje takie $k_i \in N$, że

$$de - 1 = k_i \cdot (p_i - 1) \qquad (**)$$

2. Rozważmy najpierw przypadek, gdy $NWD(a,n) = 1$ co jest równoważne temu, że dla każdego i=1,2,...,k, p_i nie dzieli a wówczas z małego twierdzenia Fermata mamy dla każdego i=1,2,...,k, $a^{p_i - 1} \equiv 1 \pmod{p_i}$ i dalej

$$a^{de-1} = a^{k_i(p_i-1)} = (a^{p_i-1})^{k_i} \equiv 1 \pmod{p_i}$$

bo kongruencje względem tego samego modułu możemy mnożyć stronami. Zatem

$$a^{de} \equiv a \pmod{p_i}$$

3. Rozważmy teraz przypadek, gdy dla każdego i=1,2,...,k p_i dzieli a wówczas oczywiście $a^{de} \equiv a \pmod{p_i}$ dla każdego i=1,2,...,k i z lematu dostajemy

$$a^{de} \equiv a \pmod{p_1 p_2 \cdot \ldots \cdot p_k}$$

co daje tezę w rozważanym przypadku. Kończąc rozważmy przypadek trzeci gdy część z modułów dzieli a a część nie dzieli a. Rozważmy tylko przypadek gdy dla i=1,2,...,r p_i nie dzieli *a a dla i=r+1,...,k,* p_i dzieli *a,* w pozostałych przypadkach rozumuje się analogicznie.

4. Z poprzednich dwóch przypadków mamy

$$a^{de} \equiv a \pmod{p_1 p_2 \cdot \ldots \cdot p_r}$$

$$a^{de} \equiv a \pmod{p_r p_{r+1} \cdot \ldots \cdot p_k}$$

i korzystając z wyników zadania xx dostajemy $a^{de} \equiv a \pmod{p_1 p_2 \cdot \ldots \cdot p_k}$

Uwaga 1. Użyteczność powyższego twierdzenia dla kryptografii polega na tym, że przy pewnych warunkach narzuconych na liczby *n, d, e*, funkcje potęgowe w pierścieniu Z_n zdefiniowane jako $Z_n \ni x \to x^d \in Z_n$ oraz $Z_n \ni x \to x^e \in Z_n$ są względem siebie odwrotne.

Uwaga 2. Warto zwrócić uwagę na to, że nie wymagamy w założeniach powyższego twierdzenia by NWD(a,n)=1 tak jak w założeniach twierdzenia Eulera .■

Zadanie 5
Pokazać, że przy oznaczeniach i założeniach zadania 8.4 warunek $de-1$ jest podzielne przez p_i-1 dla każdego $i\in<1,k>$ wynika bezpośrednio z warunku

$$de\equiv 1 \pmod{\varphi(\mathrm{n})}.$$

dla dowolnego $n\in N, n\geq 5$.

Rozwiązanie
1. Istotnie z własności funkcji Eulera wynika, że $\varphi(n)=(p_1-1)(p_2-1)...(p_k-1)$, zatem dla pewnej liczby całkowitej l mamy $de-1=l\cdot(p_1-1)(p_2-1)...(p_k-1)=l\cdot\varphi(n)$ a więc $de-1$ jest podzielne przez p_i-1 dla każdego $i\in<1,k>$.■

Zadanie 6
Pokazać, że w twierdzeniu o poprawności szyfru RSA założenie bezkwadratowości liczby *n* jest istotne.

Rozwiązanie
1. Bezkwadratowość liczby *n* w dowodzie twierdzenia o poprawności szyfru RSA jest potrzebna w momencie korzystania z lematu będącego treścią zadania xx. (Z tego, że $a\equiv b(\mathrm{mod}\, p)$ nie wynika, że $a\equiv b(\mathrm{mod}\, p^2)$. Być może, że istnieje jednak dowód nie korzystający z tego lematu.

2. Niech $n=p^2$, gdzie *p*=5 mamy wówczas $\varphi(n)=\varphi(25)=5(5-1)=20$. W pierścieniu Z_{20} liczby $e=3$ i $d=7$ są wzajemnie odwrotne, mamy więc dla *a* =2

$$(a^d)^e(\mathrm{mod}\, p^2)=(2^7)^3(\mathrm{mod}\, 25)\equiv 3^3(\mathrm{mod}\, 25)=9.$$

Ponieważ $9\neq 2(\mathrm{mod}\, 25)$ teza twierdzenia o poprawności szyfru RSA nie zachodzi, co dowodzi istotności założenia o bezkwadratowości liczby *n*. ■

Zadanie 7
Niech będzie dany system kryptograficzny ElGamala zdefiniowany dla ciała F_q z przyjętym, ustalonym generatorem $g\in F_q^*$ grupy multiplikatywnej F_q^*.

Pokazać, że jeśli dysponujemy szyfrogramem $(g^k, m\otimes_q (g^a))^k)$ (gdzie $m\in F_q^*$ jest tekstem jawnym, $a\in<2,q-2>$ tajnym kluczem prywatnym, g^a kluczem publicznym oraz *k* jest realizacją zmiennej losowej dyskretnej $X:\Omega\to<2,q-2>$ określonej na pewnej przestrzeni probabilistycznej $(\Omega,\mathfrak{M},P)$) to znając tajny klucz prywatny *a* potrafimy odtworzyć z szyfrogramu $(g^k, m\otimes_q (g^a)^k)$ wiadomość jawną *m*.

Rozwiązanie

Jeśli znamy tajny klucz prywatny a , to możemy łatwo obliczyć a –tą potęgę elementu g^k w ciele F_q (pierwsza współrzędna szyfrogramu) otrzymując $(g^k)^a = g^{ka}$. Stosując rozszerzony algorytm Euklidesa obliczamy $(g^{ka})^{-1} = g^{-ka}$. Wykorzystując, obliczamy element g^{-ka}, można teraz z drugiej współrzędnej szyfrogramu czyli $m \otimes_q (g^a)^k$ odtworzyć wiadomość jawną. Istotnie

$$(m \otimes_q (g^a)^k) \otimes_q g^{-ka} = (m \otimes_q g^{ka}) \otimes_q g^{-ka} = m \otimes_q (g^{ka} \otimes_q g^{-ka}) = m \otimes_q 1 = m .$$

Uwaga 1. Element $g \in F_q^*$ nie musi być generatorem grupy multiplikatywnej F_q^*. Nigdzie w dowodzie nie wykorzystujemy tego faktu. Warto jednak z pewnych względów jako g wybrać generator grupy F_q^*.

Uwaga 2. W sytuacji, gdy potrafimy obliczyć logarytm dyskretny $\log_g x$ w grupie multiplikatywnej F_q^*, (gdzie x jest kluczem publicznym g^a) system kryptograficzny ElGamala dla danych parametrów q, g i a jest złamany, bo z klucza publicznego g^a potrafimy obliczyć tajny klucz prywatny a.

Uwaga 3. Z podobnych przyczyn nie należy wybierać jako klucza tajnego wartości $a = 1$ i $a = q - 1$ ponieważ w pierwszym przypadku $g^1 = g$ a w drugim $g^{q-1} = 1$. Zatem z wartości klucza publicznego i wartości g jesteśmy w stanie łatwo wywnioskować jaką wartość ma klucz tajny a. ■

Zadanie 8

Uzasadnić dlaczego w systemie kryptograficznym ElGamala jest lepiej, jeśli element $g \in F_q^*$ definiujący kryptosystem jest generatorem grupy multiplikatywnej F_q^*.

Rozwiązanie

1. Załóżmy, że $g \in F_q^*$ jest generatorem grupy multiplikatywnej F_q^*, wówczas funkcja $f :< 2, q-2 >\ni k \to g^k \in F_q^*$ jest różnowartościowa i przyjmuje wartości ze zbioru $A = F_q^* \setminus \{1, g\}$. Jeśli zmienna losowa X wybierająca elementy ze zbioru $< 2, q-2 >$ ma rozkład jednostajny, to zmienna losowa g^X ma rozkład jednostajny na zbiorze *A*.

2. Jeśli $g \in F_q^*$ nie jest generatorem grupy multiplikatywnej F_q^*, to zbiór $f(< 2, q-2 >)$ jest istotnie mniejszy (rząd elementu grupy jest dzielnikiem rzędu grupy) od zbioru $F_q^* \setminus \{1, g\}$. Zatem zmienne losowe g^X i $P \otimes_q (g^a)^X = P \otimes_q (g^X)^a$ przyjmują wartości z mniejszego zbioru. Intuicyjnie rzecz biorąc, mamy więc „mniejszą losowość" niż w przypadku, gdy *g* jest generatorem i to stanowi argument przemawiający za wyborem jako g generatora F_q^*.

3. Warto również zauważyć, że w przypadku, gdy g nie jest generatorem F_q^* to jeden klucz publiczny g^a może mieć wiele kluczy tajnych prywatnych (por. zad. xx). Może to być zarówno zaletą jak i wadą. ■

Zadanie 9

Pokazać, że w twierdzeniu o poprawności szyfru RSA, założenie bezkwadratowości liczby *n,* jest istotne.

Rozwiązanie

1. Bezkwadratowość liczby *n* w dowodzie twierdzenia o poprawności szyfru RSA jest potrzebna w momencie korzystania z lematu będącego treścią zadania xx (z tego , że $a \equiv b(\mathrm{mod}\, p)$ nie wynika, że $a \equiv b(\mathrm{mod}\, p^2)$). Być może, że istnieje jednak dowód nie korzystający z tego lematu.

2. Niech $n = p^2$, gdzie $p = 5$ mamy wówczas $\varphi(n) = \varphi(25) = 5(5-1) = 20$. W pierścieniu Z_{20} liczby $e = 3$ i $d = 7$ są wzajemnie odwrotne, mamy dla $a = 2$ $(a^d)^e (\mathrm{mod}\, p^2) = (2^7)^3 (\mathrm{mod}\, 25) \equiv 3^3 (\mathrm{mod}\, 25) = 9$. Ponieważ $9 \neq 2 (\mathrm{mod}\, 25)$ teza twierdzenia o poprawności szyfru RSA nie zachodzi, co dowodzi istotności założenia o bezkwadratowości liczby *n*. ■

Zadanie 10

Pokazać, że w systemie kryptograficznym ElGamala wartości k=1 i k=q-1 losowego parametru k (por. zadanie xx) nie zapewniają utajnienia szyfrogramu.

Rozwiązanie

1. Istotnie jeśli $g \in F_q^*$, gdzie F_q jest ciałem, w którym przeprowadzamy obliczenia, to dla k=1 dostajemy następującą postać szyfrogramu

$$\left(g^1, P \otimes_q \left(g^a \right)^1 \right) \qquad (*)$$

2. Jeśli g jest generatorem grupy multiplikatywnej F_q^*, to patrząc na pierwszą współrzędną szyfrogramu (*), stwierdzamy, że k musi równać się 1 (g jest ogólnie znane). Wystarczy więc drugą współrzędną szyfrogramu pomnożyć przez odwrotność klucza publicznego g^a, by odtworzyć wiadomość jawną P bez znajomości tajnego klucza prywatnego a.

3.Jeśli g nie jest generatorem grupy multiplikatywnej F_q^*, to sytuacja jest nieco bardziej złożona. Nie jesteśmy pewni czy k=1, wiemy natomiast na pewno, że $k \in I$, gdzie

$$I \overset{df}{=} \{x \in <1, q-1>; \quad x = j \cdot rz\ g + 1, \quad j \in N\}$$

gdzie $rz\ g$ oznacza rząd elementu g. Musimy wtedy, by zdefiniować szyfrogram, mnożyć drugą współrzędną szyfrogramu przez g^{-ak} dla kolejnych $k \in I$.

4.Jeśli wylosowaliśmy k=q-1 i $g \in F_q^*$ jest generatorem grupy multiplikatywnej F_q^*, to mamy $g^k = 1$. Zatem na pierwszej współrzędnej szyfrogramu pojawia się 1, a druga współrzędna zawiera po prostu tekst jawny. Istotnie

$$P \otimes_q \left(g^a \right)^k = P \otimes_q \left(g^a \right)^{q-1} = P \otimes_q \left(g^{q-1} \right)^a = P \otimes_q 1 = P$$

5. Jeśli wylosowaliśmy k=q-1 i $g \in F_q^*$ nie jest generatorem grupy multiplikatywnej F_q^*, to (ponieważ rząd elementu grupy jest dzielnikiem rzędu grupy) mamy również g^k=1 i na pierwszej współrzędnej szyfrogramu pojawia się 1. Nie wiemy jednak w tym przypadku, jakie rzeczywiście wylosowane zostało k, bo taki sam wynik da każde $k = j \cdot rz\ g$, *dla* $j \in N$. W celu deszyfracji szyfrogramu musimy mnożyć drugą współrzędną szyfrogramu przez $\left(g^{-a}\right)^k$; czyli g^{-ak} dla kolejnych $k = j \cdot rz\ g$, gdzie j=1,2,...,(q-1)/$rz\ g$.

6. Reasumując jeśli *g* jest generatorem grupy multiplikatywnej F_q^*, to tekst jawny praktycznie nie jest zaszyfrowany. Jeśli *g* nie jest generatorem, to kryptoanaliza jest znacznie ułatwiona. Nie należy więc przyjmować *k*=1 i *k*=*q*-1 . □

Zadanie 11

Niech będzie dany system kryptograficzny ElGamala zdefiniowany dla pewnego ciała F_q tak jak w zadaniu xx. Znaleźć rozkład prawdopodobieństwa zmiennej losowej $Y_1 = g^X$ (pierwsza współrzędna kryptogramu) i zmiennej losowej $Y_2 = P \otimes_2 (g^a)^X$ druga współrzędna kryptogramu. Zakładamy, że zmienna losowa dyskretna $X : \Omega \to <1, q-1>$ opisująca losowanie parametru *k* jest określona na pewnej przestrzeni probabilistycznej $(\Omega, \mathfrak{M}, P)$ i ma rozkład równomierny tzn. dla każdego $i, j \in <1, q-1>$, $i \neq j$ mamy $P(X = i) = P(X = j)$.

Rozwiązanie

1. Jeśli $g \in F_q^*$ jest generatorem grupy multiplikatywnej F_q^* oraz zmienna losowa *X* ma rozkład równomierny na $<1, q-1>$, to zmienna losowa $Y_1 \overset{df}{=} g^X$ ma rozkład równomierny na F_q^* ponieważ funkcja $<1, q-1> \ni r \to g^r \in F_q^*$ jest równowartościowa i „na”. Jeśli *g* nie jest generatorem grupy multiplikatywnej F_q^*, to $Y_1 = g^X$ ma rozkład równomierny na $\{g^r ; r \in <1, rz\ g>\}$, gdzie . $rz\ g$ oznacza rząd elementu *g*.

2. Rozważmy teraz zmienną losową $Y_2 = P \otimes_2 (g^a)^X$ dla dowolnego ustalonego tekstu jawnego $P \in F_q^*$. Musimy tak jak poprzednio rozważyć dwa przypadki, gdy g^a jest generatorem grupy multiplikatywnej F_q^* (a tak jest, jeśli *g* jest generatorem F_q^* i $\mathrm{NWD}(a, q-1) = 1$) i drugi przypadek, gdy g^a nie jest generatorem F_q^*. W pierwszym przypadku $(g^a)^X$ ma rozkład jednostajny na F_q^*, a w drugim jednostajny na zbiorze $A_1 = \{g^{a \cdot r} ; r \in <1, rz\ g^a>\} \subset F_q^*$. Mnożenie przez $P \in F_q^*$ tzn. odwzorowanie

$$F_q^* \ni x \to P \otimes_q x \in F_q^*$$

jest równowartościowe i „na”. Zatem jeśli g^a jest generatorem grupy multiplikatywnej F_q^*, to rozkład zmiennej losowej Y_2 jest jednostajny na F_q^*. W drugim przypadku jednostajny na zbiorze

$$A_2 = \{P \otimes_q g^{ar} ; r \in <1, rz\ g^a>\}$$

Uwaga. Wartość k=1 i k=q-1 albo nie zapewniają tajności szyfrogramu albo znacznie ułatwiają kryptoanalizę. W przypadku więc, gdy zmienna losowa X przyjmie jedną z tych wartości, bezpieczniej jest ponowić losowanie. Jeśli pomijamy *1* i *q-1,* i przyjmujemy, że zmienna losowa X ma rozkład równomierny na $<2,q-2>$ i g jest generatorem F_q^*, to zmienna losowa $Y_1 = g^X$ ma rozkład równomierny na $F_q^* \setminus \{1, g\}$, ponieważ funkcja $<2, q-2> \ni r \to g^r$ jest różnowartościowa i "na". Jeśli g nie jest generatorem grupy multiplikatywnej F_q^*, to $Y_1 = g^X$ ma rozkład równomierny na zbiorze $\{g^r ; r \in <2, rz\ g>\}$. ■

Zadanie 12
Zaproponować system kryptograficzny z kluczem publicznym z wieloma istotnie różnymi kluczami prywatnymi i jednym kluczem publicznym. Wskazówka. Wykorzystać system kryptograficzny ElGamala.

Rozwiązanie
Przyjmijmy oznaczenia jak w zadaniu xx. Wystarczy wybrać w systemie kryptograficznym ElGamala takie q (gdzie q jest ilością elementów ciała F_q), by liczba q-1 (ilość elementów grupy multiplikatywnej F_q^*) nie dawała się tak łatwo rozłożyć na czynniki pierwsze. Jeśli teraz g nie jest generatorem grupy multiplikatywnej F_q^* oraz $a \in <2, q-2>$ jest ustalonym kluczem prywatnym, to również każda liczba

$$a_j = [a + j \cdot rz\ g]_{q-1} \qquad (*)$$

dla $j \in <1, rz\ g>$, (gdzie $rz\ g$ jest rzędem elementu g) będzie dobrym tajnym kluczem prywatnym odpowiadającym kluczowi publicznemu g^a.

Istotnie, każdej z liczb a_j dla $j \in <1, rz\ g>$ potrafimy zdeszyfrować kryptogram $(g^k, P \otimes_q (g^a)^k)$, ponieważ deszyfracja sprowadza się do obliczenia g^{ka}, a g^{ka} liczymy mając a_j w sposób następujący

$$g^{ka_j} = g^{k[a+j\cdot rz\ g]_{q-1}} = g^{k(a+j\cdot rz\ g)} = g^{ka} \cdot g^{j\cdot rz\ g} = g^{ka} \quad ■$$

Zadanie 13
Konstruujemy pewien szyfr z kluczem publicznym w następujący sposób. Wybieramy dwie liczby całkowite *a ,b* oraz wyliczamy *M=ab*-1, wybieramy następnie dwie liczby całkowite *a`* i *b`*, oraz przyjmujemy:

$$e = a'M + a \qquad d = b'M + b$$

a ponadto obliczamy:

$$n = \frac{ed-1}{M} = a'b'M + ab' + a'b + 1$$

Kluczem publicznym jest para uporządkowana (*n, e*), a kluczem prywatnym para uporządkowana (*n, d*). Szyfrowanie polega na obliczeniu

$$c = e \cdot m (\mathrm{mod}\, n),$$

gdzie $m \in N \cup \{0\}$ jest wiadomością jawną. Deszyfrowanie polega na obliczeniu

$$c \cdot d (\mathrm{mod}\, n).$$

1) Sprawdzić, czy po deszyfracji otrzymamy wiadomość jawną.
2) Pokazać, jak skonstruować podpis cyfrowy.
3) Pokazać, w jaki sposób za pomocą algorytmu Euklidesa można złamać powyższy szyfr.

Rozwiązanie
1. Sprawdzenie tego, czy po deszyfracji kluczem *d* kryptogramu *c*, otrzymamy wiadomość jawną *m*, sprowadza się do wykazania, że:

$$(e \cdot m (\mathrm{mod}\, n)) \cdot d (\mathrm{mod}\, n) = m \qquad (1)$$

Korzystając z własności homomorfizmu jakim jest branie reszty z dzielenia przez *n*, dostajemy równoważną postaci (1) – postać (2).

$$e \cdot m \cdot d (\mathrm{mod}\, n) = m \qquad (2)$$

Wstawiając definicje kluczy *e* i *d* do (2), dostajemy:

$$(a`M + a)(b`M + b) \cdot m (\mathrm{mod}\, n) = m \qquad (3)$$

czyli:

$$((a`M + a)(b`M + b) \cdot m - m)(\mathrm{mod}\, n) = 0$$

$$m((a`M + a)(b`M + b) - 1)(\mathrm{mod}\, n) = 0$$

$$m \cdot M \cdot n(\mathrm{mod}\, n) = 0$$

ale ostatnia równość jest oczywista.

2. Za podpis cyfrowy dokumentu *m* można przyjąć parę uporządkowaną $(m, d \cdot m(\mathrm{mod}\, n))$, gdzie $n = \frac{ed-1}{M} = a'b'M + ab' + a'b + 1$, a (*n, d*) jest kluczem prywatnym. Sprawdzanie

podpisu sprowadza się w tej sytuacji do sprawdzenia, czy:

$$e \cdot d \cdot m(\mathrm{mod}\, n) = m \qquad (4)$$

jeśli równość (4) jest spełniona, akceptujemy podpis dokumentu *m*.

3. Zauważmy, że za pomocą algorytmu Euklidesa (dokładniej: rozszerzonego algorytmu Euklidesa) łatwo można obliczyć element odwrotny do *e* (jeśli taki element istnieje tzn. jeśli $NWD(e, n) = 1$). Widać, że tym elementem odwrotnym jest *d*, ponieważ dla *m*=1 mamy:

$$ed(\mathrm{mod}\, n) = 1$$

$$de(\mathrm{mod}\, n) = 1$$

a zatem: $e^{-1} = d$.

Uwaga. Konstruowanie kryptosystemu z kluczem publicznym bazującego na mnożeniu wiadomości jawnej a następnie uzyskanego kryptogramu przez elementy względem siebie odwrotne (odwrotne w pewnym pierścieniu Z_n) jest naturalnym prostym pomysłem. System taki z uwagi na istniejące szybkie algorytmy odwracania elementu pierścienia nie jest jednak bezpieczny. Zaletą przedstawionego kryptosystemu polega na tym, że najpierw wybieramy klucz szyfrujący *e* i deszyfrujący *d a* następnie do nich dopasowujemy *n*. Unikamy w ten sposób obliczania elementu odwrotnego w dużym pierścieniu. ■

Zadanie 14
Pokazać, że w systemie kryptograficznym RSA rozkład liczby *n* na czynniki pierwsze łamie RSA.