

Wykład 1. i 2. Wprowadzenie do kryptografii

Wykładowca: Stefan Dziembowski

Skryba: Stefan Dziembowski

Streszczenie. Wprowadzamy podstawowe pojęcia kryptograficzne: schematy szyfrowania (w modelu symetrycznym i asymetrycznym), schematy uwierzytelniania i podpisywania, funkcje jednokierunkowe i funkcje jednokierunkowe z zapadką. Przypominamy niektóre pojęcia teorii złożoności, teorii obliczeń, teorii liczb, algebry i rachunku prawdopodobieństwa. Większość definicji sformułowana jest w sposób nieformalny, a fakty i twierdzenia podawane są bez dowodów.

1 Wstęp

Na dzisiejszym wykładzie przedstawimy podstawowe pojęcia kryptograficzne. Dla oszczędzenia czasu większość definicji podana będzie w sposób nieformalny. Formalne definicje znajdują się np. w [2].

2 Narzędzia

Zakładamy, że uczestnicy wykładu są oswojeni z podstawowymi pojęciami rachunku prawdopodobieństwa, teorii obliczeń i teorii złożoności. W szczególności nieobce jest im pojęcie *maszyny Turinga* (deterministycznej lub nie) oraz klas P, NP, oraz PSPACE.

Mówimy, że maszyna Turinga jest *zrandomizowana* (lub. probabilistyczna) jeśli posiada dodatkową taśmę w której każdej komórce zapisano losowy bit (ze zbioru $\{0, 1\}$)¹.

Przez *protokół* będziemy rozumieć skończony ciąg (długości większej niż 1) *interaktywnych maszyn Turinga*. Pojęcia *maszyna Turinga* będziemy używać wymiennie z pojęciem *algorytmu*. W szczególności często będziemy używać pojęcia *Wielomianowego Algorytmu Probabilistycznego (WAP)*. Innymi modelami, które będziemy rozważać są *obowody logiczne* (ang.: *Boolean circuit*) oraz *maszyny RAM*.

Funkcja $\alpha : \mathcal{N} \rightarrow \mathcal{R}$ jest *zaniedbywalna* (ang.: *negligible*) jeśli jej wartość absolutna maleje szybciej niż odwrotność dowolnego wielomianu. Inaczej mówiąc: dla dowolnego $c > 0$ istnieje takie n_0 , że dla każdego $n > n_0$ mamy $|\alpha(n)| < n^{-c}$.

Jeśli mamy do czynienia z ciągiem zdarzeń losowych E_1, E_2, \dots , to mówimy, że zdarzenia te zachodzą z *zaniedbywalnym prawdopodobieństwem*, jeśli funkcja $\alpha(i) = P[E_i]$ jest zaniedbywalna. Jeśli zdarzenia $\neg E_1, \neg E_2, \dots$ są zaniedbywalne, to mówimy, że E_1, E_2, \dots zachodzą z *przeważającym prawdopodobieństwem* (ang.: *overwhelming probability*).

¹W tego typu kontekstach pisząc *losowy* zakładamy, że każdy element ma jednakowe prawdopodobieństwo wybrania.

3 Szyfrowanie

Co to jest bezpieczny *schemat szyfrowania* (ang.: *encryption scheme*), inaczej: *szyfr* (ang.: *cipher*) *kryptosystem* (ang.: *cryptosystem*)? Nie jest to pytanie na które istnieje jedna prosta odpowiedź. Aż do lat 80tych ubiegłego wieku zajmowano się szyfrowaniem bez wprowadzania formalnych definicji. Najpierw zajmiemy się zdefiniowaniem pojęcia sposobu działania schematu (jest to w miarę łatwe). Potem powiemy co oznacza jego *bezpieczeństwo*.

3.1 Definicja kryptosystemu (symetrycznego)

Symetryczny schemat szyfrowania składa się z trzech następujących algorytmów (maszyn Turinga): algorytm *generacji klucza* G , algorytm *szyfrujący* E i algorytm *odszyfrowujący* D . Algorytm G nie pobiera wejścia i zwraca *klucz* K (ze skończonego zbioru kluczy $\mathcal{K} \subset \{0, 1\}^*$). Algorytm E pobiera na wejściu klucz K i *tekst jawny* (*wiadomość*, (ang.: plaintext)) M (ze zbioru wiadomości $\mathcal{M} \subseteq \{0, 1\}^*$). Algorytm E zwraca *szyfrogram* (zwany też *kryptogramem*):

$$C = E(K, M) = E_K(M).$$

Algorytm D pobiera na wejściu klucz K i szyfrogram C i zwraca

$$D(K, C) = D_K(C).$$

Algorytmy G , E i D mogą być zrandomizowane i posiadać stan (np. licznik). Często zamiast specyfikować G podaje się opis zbioru kluczy \mathcal{K} . Wówczas zakładamy, że K jest wybrany losowo ze zbioru \mathcal{K} . Wymagamy, by zachodziło

$$\forall K \in \mathcal{K} \forall M \in \mathcal{M} D_K(E_K(M)) = M.$$

Ponadto schemat szyfrowania powinien być *bezpieczny*.

3.2 Definicja bezpieczeństwa

Nie ma jednej definicji bezpieczeństwa. Dany szyfr może być bezpieczny w pewnych zastosowaniach i nie nadawać się do innych celów. Tradycyjnie rozważa się scenariusz z trzema osobami (zwanymi też: stronami, graczami, partiami, etc.):

- *użytkownikami* — Alicją i Bobem oraz
- *przeciwnikiem* — Ewą.

Zakładamy, że Alicja i Bob ustalili tajny (czyli znany tylko im) losowy klucz $K \in \mathcal{K}$. Wszystkie osoby (w tym Ewa!) znają schemat szyfrowania (tj. algorytmy E i D). Jest to tzw. *zasada Kirkhoffsa*. Ogólnie, w kryptografii przyjmujemy (pesymistyczne) założenie, że wszystkie dane, które nie są explicite zdefiniowane jako tajne, są znane przeciwnikowi.

Alicja i Bob wysyłają do siebie wiadomości zaszyfrowane za pomocą danego schematu szyfrowania. Do przesyłania szyfrogramów użytkownicy korzystają z niezabezpieczonego łącza. Mimo dostępu do tego łącza, Ewa nie powinna mieć żadnej

informacji o przesyłanych wiadomościach (poza, ewentualnie, ich długością). Definicja (w uproszczeniu) wygląda następująco. Przyjmijmy, że Ewa jest zrandomizowaną interaktywną maszyną Turinga. Rozważamy taką grę:

1. Ewa wybiera dwie wiadomości M_0 i M_1 (tej samej długości) i wysyła je do kogoś, kto zna klucz K . Dla ustalenia uwagi niech będzie to Alicja.
2. Alicja losuje bit $b \in \{0, 1\}$, oblicza $C = E_K(M_b)$ i wysyła C do Ewy.
3. Ewa ma zgadnąć b .

(Ogólniej: Ewa wybiera dwa *ciągi* wiadomości, Alicja odpowiada ciągami szyfrogramów.) Zauważmy: jeśli Ewa np. potrafi obliczyć pierwszy bit tekstu jawnego na podstawie szyfrogramu, to łatwo dobierze M_0 i M_1 w taki sposób, że zgadnie b w ostatnim kroku. Szyfr jest *idealnie bezpieczny* (ang.: *perfectly secure*), jeśli dowolna Ewa dysponująca nieograniczoną mocą obliczeniową zgaduje b poprawnie z prawdopodobieństwem $\frac{1}{2}$. Historycznie pierwsza definicja idealnego bezpieczeństwa sformułowana została przez Shannona w 1949 [7] za pomocą języka teorii informacji. Jest ona równoważna powyższej. Shannon pokazał również, że w każdym szyfrze idealnie bezpiecznym klucz musi być co najmniej tej samej długości co wiadomość. Ponadto, dany klucz może być użyty tylko raz (poważnie ogranicza praktyczne zastosowania takich szyfrów).

Przykładem szyfru idealnie bezpiecznego jest *szyfr Vernama* (ang.: *Vernam's cipher, One-Time Pad*). W szyfrze tym długość n klucza jest równa długości wiadomości. Szyfrowanie zdefiniowane jest następująco

$$E((K_1, \dots, K_n), (M_1, \dots, M_n)) = (K_1 \oplus M_1, \dots, K_n \oplus M_n).$$

Odszyfrowywanie jest identyczne z szyfrowaniem: $D(K, M) = C(K, M)$. Nietrudno pokazać, że szyfr Vernama jest idealnie bezpiecznym schematem szyfrowania.

W praktyce stosuje się szyfry, które nie są idealnie bezpieczne (ale za to klucz jest krótki i może być używany wielokrotnie). Nieformalnie mówiąc, kryptosystem jest bezpieczny, jeśli

- Ewa dysponująca *ograniczoną mocą obliczeniową*
- zgaduje poprawnie b z prawdopodobieństwem co najwyżej $\frac{1}{2} + \epsilon$.

Wybór ograniczenia na moc obliczeniową oraz wartości ϵ jest kwestią gustu. Inaczej mówiąc: nie ma sensu podział bezpieczny/niebezpieczny. Raczej należy mówić np.: „bezpieczny z błędem $\epsilon = 0.0000001$ względem przeciwnika o dysponującego maszyną Turinga wykonującą maksymalnie 10000000 kroków”. Wybór modelu obliczeń (maszyna Turinga, maszyna RAM, etc.) też jest kwestią wyboru. Dla uproszczenia najczęściej w rozważaniach teoretycznych wybieramy tzw. *podejście asymptotyczne*. Wprowadzamy wówczas *parametr bezpieczeństwa i* . Algorytm G oraz algorytm Ewy pobierają dodatkowe wejście i . Mówimy, że kryptosystem (E, D) jest *wielomianowo bezpieczny*, jeśli

- Ewa dysponująca czasem wielomianowym ze względu na i

- zgaduje poprawnie b z prawdopodobieństwem co najwyżej zanedbywalnie większym od $\frac{1}{2}$.

Takie podejście ma tę podstawową zaletę, że nie musimy precyzować modelu obliczeń (wszystkie są sobie równoważne ze względu na wielomianową redukcję).

3.3 Praktyczne szyfry symetryczne

Do najbardziej znanych szyfrów symetrycznych należą szyfry blokowe (np. DES, IDEA i AES) użyte w jednym z trybów (np. CBC). Szczegółowe ich sposobu działania oraz trybów użycia nie mieści się w programie wykładu. Zainteresowany odsyłamy do [2, 5].

4 Funkcje jednokierunkowe

Podstawowym problemem współczesnej kryptografii jest to, że nie znane są jakiegokolwiek praktyczne (tj. wydajne i z kluczem krótszym niż wiadomość) schematy szyfrowania których wielomianowego bezpieczeństwa potrafimy dowieść bez zakładania prawdziwości nieudowodnionych hipotez². Z istnienia wielomianowo bezpiecznego schematu szyfrowania wynika, że $P \neq NP$, co jest jednym z najbardziej znanych (i prawdopodobnie najtrudniejszych) problemów otwartych współczesnej matematyki. Niestety nie znany jest dowód implikacji odwrotnej. Pokazanie, że jakiś szyfr jest wielomianowo bezpieczny jest więc (w nieformalnym sensie) trudniejsze, niż pokazanie, że $P \neq NP$.

Nie oznacza to, że w kryptografii niczego nie da się dowieść. Możemy np. dowieść bezpieczeństwa zakładając prawdziwość wybranych hipotez. Podstawowym założeniem tego typu jest istnienie funkcji jednokierunkowych.

Definicja 1 *Funkcja $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ jest jednokierunkowa (ang.: one-way) jeśli*

- *jest obliczalna w czasie wielomianowym oraz*
- *jest trudna do odwrócenia, tzn. dla dowolnego WAP A wartość*

$$\Pr_{x \leftarrow \{0,1\}^i} [f(A(f(x))) = f(x)]$$

jest zanedbywalna jako funkcja i (zapis $x \leftarrow X$ oznacza, że x jest wybrany losowo ze zbioru X).

Powyższa definicja jest uproszczona. Pełną definicję (w rozmaitych wariantach) można znaleźć w Rozdziale 2 [2].

Zauważmy, że może się zdarzyć, że funkcja jest trudna do odwrócenia, ale na podstawie $f(x)$ można obliczyć np. pierwszą połowę x . Jest to w kontraście z wymaganiami bezpieczeństwa dla szyfrowania (gdzie umiejętność obliczenia wartości dowolnego bit tekstu jawnego pozwala Ewie wygrać).

²Nie jest to tylko problem z wielomianowym bezpieczeństwem. Ogólnie w tej dziedzinie nikt nie umie dowieść jakiegokolwiek nietrywialnego bezpieczeństwa praktycznych schematów szyfrowania, poza schematami w modelach niestandardowych (np. w modelu kwantowym).

Jeśli istnieją wydajne wielomianowo bezpieczne schematy szyfrowania, to muszą istnieć funkcje jednokierunkowe. Założenie ich istnienia jest więc dla kryptografii konieczne. Okazuje się, że jest też wystarczające.

Fakt 2 *Wydajne wielomianowo bezpieczne symetryczne schematy szyfrowania istnieją wtedy i tylko wtedy gdy istnieją funkcje jednokierunkowe.*

Zredukowano więc założenie bardziej skomplikowane (istnienie wielomianowo bezpiecznych schematów szyfrowania), do założenia prostszego (istnienie funkcji jednokierunkowych).

Z braku czasu nie podamy w tym miejscu przykładu funkcji jednokierunkowej (przykłady takie można znaleźć np. w [2]). Należy pamiętać, że przykładem takim może być dowolna funkcja jednokierunkowa z zapadką, w szczególności RSA (patrz Rozdział 6.2.2).

5 Uwierzytelnianie

W modelu rozważanym w Rozdziale 3 przeciwnik był pasywny, to znaczy jedynym jego celem było poznanie zaszyfrowanych wiadomości. W praktyce (np. w Internecie) dodatkowym zagrożeniem są aktywne ataki polegające na próbie podszycia się pod jednego z użytkowników. Konkretnie, celem Ewy może być np. wyprodukowanie wiadomości w taki sposób by po jej otrzymaniu Bob był przekonany, że pochodzi ona od Alicji³. Do zapobieżenia tego typu atakom służą *Kody Uwierzytelniające Wiadomości* (ang.: *Message Authentication Codes (MAC)*), zwane też *Schematami Uwierzytelniającymi Wiadomości*.

5.1 Definicja schematu uwierzytelniającego wiadomości

Schemat uwierzytelniający wiadomości jest trójką algorytmów: algorytm generacji klucza G (działający tak samo jak w przypadku schematów szyfrowania), algorytm *oznaczający* (T) (ang.: *tagging algorithm*) i algorytm *weryfikujący* (V). Algorytm T pobiera na wejściu *klucz* $K \in \mathcal{K}$ i *wiadomość* $M \in \mathcal{M}$. Algorytm T zwraca *oznacznik* (ang.: *tag*).

$$\sigma = T(K, M) = T_K(M).$$

Algorytm V pobiera na wejściu *klucz* K , *oznacznik* σ i *wiadomość* M i zwraca

$$V(K, M, \sigma) = V_K(M, \sigma) \in \{\text{yes}, \text{no}\},$$

Przy czym wymagamy, by

$$\forall K \in \mathcal{K} \forall M \in \mathcal{M} V_K(M, T_K(M)) = \text{yes}.$$

Nie podamy to formalnej definicji bezpieczeństwa (znajduje się ona w Rozdziale 8 [2]). W skrócie mówiąc wygląda ona podobnie do definicji bezpieczeństwa w przypadku szyfrów. Mianowicie, określamy grę w której Ewa może zapytać wyrocnie

³Uwaga: fakt, że wiadomość jest poprawnie zaszyfrowana nie daje gwarancji, że pochodzi ona od uczciwego użytkownika. Przykładem może być szyfr Vernama.

znającą klucz K o oznaczniki dowolnie wybranych przez siebie wiadomości. Następnie Ewa ma za zadanie wyprodukować parę (wiadomość M , oznacznik σ), takie, że $V_K(M, T_K(M)) = \text{yes}$. Jeśli nie istnieje Ewa (o ograniczonej mocy obliczeniowej), która radzi sobie z tym zadaniem ze znaczącym prawdopodobieństwem, to mówimy, że schemat jest bezpieczny. Ponownie, precyzyjne określenie znaczenia słów „ograniczona moc obliczeniowa” i „znaczące prawdopodobieństwo” jest kwestią wyboru. W szczególności tak jak poprzednio możemy mówić o schematach wielomianowo bezpiecznych.

5.2 Istnienie schematów uwierzytelniających wiadomość

Podobnie jak w przypadku szyfrowania, nie istnieją wydajne schematy uwierzytelniające których bezpieczeństwa potrafimy dowieść. Możemy za to pokazać następujący fakt.

Fakt 3 *Wydajne schematy uwierzytelniające wiadomość wielomianowo bezpieczne istnieją wtedy i tylko wtedy gdy istnieją funkcje jednokierunkowe.*

5.3 Praktyczne schematy uwierzytelniające wiadomość

Podobnie jak w przypadku szyfrów symetrycznych nie będziemy tu omawiać praktycznych implementacji. Zainteresowany odsyłamy do [2, 5].

6 Kryptografia klucza publicznego (asymetryczna)

Do tego momentu omawialiśmy kryptografię *symetryczną*. Jej główną cechą jest to, że ten sam klucz służy do szyfrowywania i odszyfrowywania (lub do oznaczania i weryfikacji). W latach 70tych rewolucję w kryptografii spowodowało wprowadzenie *kryptografii asymetrycznej* (inaczej: *kryptografii klucza publicznego*) [1, 6]. Mamy w niej do czynienia z parą kluczy: (*klucz publiczny e , klucz prywatny d*), przy czym znajomość klucza publicznego nie ułatwia (w sposób znaczący) obliczenia wartości klucza prywatnego. W przypadku *asymetrycznych schematów szyfrowania*

- klucz publiczny e służy do szyfrowania, zaś
- klucz prywatny d służy do odszyfrowywania

W przypadku asymetrycznych schematów uwierzytelniania (zwanych w tym kontekście *schematami podpisywania*)

- klucz publiczny e służy do weryfikacji, zaś
- klucz prywatny d służy do oznaczania zwanego tu: *podpisywaniem*; algorytm generowania podpisu oznaczamy przez S (zamiast T).

Poważną wadą kryptografii asymetrycznej jest jej niska wydajność. Podstawową zaletą jest to, że klucz publiczny (jak sama nazwa wskazuje) można ogłosić publicznie. W przypadku schematów szyfrowania oznacza to, że, aby w sposób tajny wysłać wiadomość do Alicji, nie musimy wcześniej uzgadniać z nią klucza (co jest konieczne w

przypadku kryptografii symetrycznej). W przypadku schematów podpisywania daje to każdemu (kto zna klucz publiczny Alicji) możliwość weryfikacji jej podpisu. Zważmy, że w przypadku symetrycznego uwierzytelniania Bob nie jest w stanie nikomu udowodnić, że dana wiadomość pochodzi od Alicji, nawet gdy posiada oznacznik wyprodukowany przez Alicję dla tej wiadomości. Jest tak dlatego, że po pierwsze – klucz jest tajny, po drugie – Bob sam jest w stanie wyprodukować taki oznacznik.

Pomniemy tu pełną definicję bezpieczeństwa (zainteresowani mogą ją znaleźć w Rozdziałach 7 i 9 [2]). W skrócie: w definicji tej rozważamy grę podobną do tej w definicji bezpieczeństwa szyfrów symetrycznych (Rozdział 3.2), z tą różnicą, że zakładamy, że Ewa zna klucz publiczny.

6.1 Parę faktów z teorii liczb

Aby pokazać przykład kryptosystemu asymetrycznego potrzebujemy krótkiej powtórki z teorii liczb. Zakładamy przy tym, że uczestnikom wykładu znane jest pojęcie grupy.

Fakt 4 *Istnieją wydajne metody sprawdzania czy dana liczba jest pierwsza.*

Najbardziej wydajnym testem na pierwszość jest zrandomizowany algorytm Millera-Rabina (znajdują się one np. [2], Rozdział C). Ma on niezerowe (choć zaniedbywalne) prawdopodobieństwo błędu. Niedawno pojawił się deterministyczny algorytm [4]. Do celów praktycznych jest on nieprzydatny.

Hipoteza 5 *Faktoryzacja iloczynów dużych liczb pierwszych jest trudna obliczeniowo, formalnie mówiąc dla dowolnego WAP A*

$$P[A(pq) = p]$$

(gdzie p i q są losowymi liczbami pierwszymi długości i), jest zaniedbywalne (jako funkcja i).

Fakt 6 *Dla dowolnych całkowitych a, b istnieją takie całkowite a' i b' , że $a'a + b'b = \gcd(a, b)$. Co więcej a' i b' można wydajnie policzyć (uogólnionym algorytmem Euklidesa).*

Dla $n \in \mathcal{N}$ symbol Z_n oznacza zbiór $0, \dots, n-1$, zaś symbol Z_n^* oznacza zbiór tych elementów z Z_n , które są względnie pierwsze z n . Moc tego zbioru jest równa $\varphi(n)$, gdzie n jest funkcją Eulera. Oczywiście jeśli p jest pierwsza, to $Z_p = Z_n \setminus \{0\}$.

Fakt 7 *Dla dowolnego $n \in \mathcal{N}$ zbiór Z_n^* jest grupą (abelową) z mnożeniem modulo n .*

Fakt 8 *Dla dowolnego $n \in \mathcal{N}$ i dowolnego a takiego, że $\gcd(n, a) = 1$ mamy $a^{|Z_n^*|} = 1$.*

(Jeśli n jest liczbą pierwszą, to powyższy fakt nosi nazwę *Małego Twierdzenia Fermata*.) Z Faktu 6 dostajemy:

Fakt 9 *Istnieje wydajny algorytm znajdujący odwrotności w Z_n^* .*

Fakt 10 *Liczby w Z_n^* można wydajnie podnosić do potęgi. Konkretnie istnieje algorytm podnoszący a do potęgi b modulo n , wydajny ze względu na sumaryczną długość a, b i n .*

Definicja 11 Niech G i H będą grupami z operacją \cdot . Funkcję $f : G \rightarrow H$ nazywamy homomorfizmem jeśli dla dowolnych $a, b \in G$ mamy $f(a \cdot b) = f(a) \cdot f(b)$. Grupy G i H są izomorficzne, jeśli istnieje między nimi homomorfizm, który jest bijekcją.

(Intuicyjnie: grupy izomorficzne mają identyczną strukturę.) Dla dowolnych grup G i H (z operacją \cdot i elementami neutralnymi 1_G i 1_H , odpowiednio), przez $G \times H$ oznaczamy grupę

- z nośnikiem równym iloczynowi kartezjańskiemu nośników G i H
- z operacją \cdot zdefiniowaną przez operacje grupy G i H wzięte po przekątnych, to znaczy

$$(g, h) \cdot (g', h') = (g \cdot g', h \cdot h')$$

- z elementem neutralnym równym $(1_G, 1_H)$.

Twierdzenie 12 (Chińskie twierdzenie o resztach) Niech m_1, \dots, m_k będą parami względnie pierwsze. Niech $m = m_1 \cdot \dots \cdot m_k$. Wówczas

1. grupy Z_m i $Z_{m_1} \times \dots \times Z_{m_k}$ są izomorficzne.
2. Podobnie grupy Z_m^* i $Z_{m_1}^* \times \dots \times Z_{m_k}^*$.

Izomorfizm ten jest zdefiniowany przez

$$f(n) := (n \bmod m_1, \dots, n \bmod m_k)$$

Jest on wydajnie obliczalny w obie strony.

6.2 RSA

W tym rozdziale opisujemy sposób działania schematu szyfrowania i podpisywania o nazwie RSA [6]. Niech i będzie ustaloną liczbą naturalną (parametrem bezpieczeństwa).

6.2.1 Wersja uproszczona

Schemat szyfrowania zdefiniowany jest następująco:

Generacja klucza Losujemy dwie liczby pierwsze p i q długości i , takie że $p \neq q$. Niech n (nazywane *modułem RSA*) będzie równe $p \cdot q$.

Bezpieczeństwo szyfru opiera się na trudności rozkładu dużych liczb naturalnych na czynniki pierwsze (Fakt 5). Będziemy działać w grupie Z_n^* . Funkcja szyfrująca którą zdefiniujemy będzie permutacją na Z_n^* , (tzn. bijekcją $Z_n^* \rightarrow Z_n^*$). Zarówno szyfrogramy jak i teksty jawne muszą być elementami Z_n^* . (Mamy $|Z_n^*| = (p-1)(q-1)$ zatem szansa, że losowa liczba jest elementem Z_n^* jest zaniedbywalna.)

Losujemy $e \in Z_{\phi(n)}^*$, tzn. takie, że $\gcd(e, \phi(n)) = 1$ i znajdujemy (uogólnionym algorytmem Euklidesa) d takie, że $ed = 1 \bmod \phi(n)$,

Kluczem publicznym jest (n, e) . Kluczem prywatnym jest (n, d) .

Funkcja szyfrująca

$$E_{n,e}^{\text{RSA}}(x) = x^e \pmod{n}.$$

Funkcja odszyfrowująca

$$D_{n,d}^{\text{RSA}}(x) = x^d \pmod{n}$$

Łatwo sprawdzić, że dla dowolnego $x \in Z_n^*$

$$D_{n,d}^{\text{RSA}}(E_{n,e}^{\text{RSA}}(x)) = (x^e)^d = x^{ed} = x \pmod{n}.$$

Schemat podpisu zdefiniowany jest następująco: **generacja klucza** jest identyczna jak w przypadku schematu szyfrowania, ponadto **funkcja weryfikująca** dana jest wzorem:

$$V_{n,e}^{\text{RSA}}(x) = x^e \pmod{n} \quad (= E_{n,e}^{\text{RSA}}(x))$$

a **funkcja podpisująca**:

$$S_{n,d}^{\text{RSA}}(x) = x^d \pmod{n} \quad (= D_{n,d}^{\text{RSA}}(x)).$$

Fakt 13 *Jeśli faktoryzacja iloczynów dużych liczb pierwszych jest łatwa (czyli Hipoteza 5 nie jest prawdziwa), to RSA nie jest bezpieczne.*

Uwaga: nie znany jest dowód implikacji w drugą stronę. Natomiast można pokazać, że:

Fakt 14 *Jeśli faktoryzacja iloczynów dużych liczb pierwszych jest trudna obliczeniowo, to trudne jest też obliczenie e na podstawie (n, e) .*

Dowód przebiega na zasadzie redukcji: pokazujemy, że mając dany wydajny algorytm obliczenia e na podstawie (n, e) potrafimy skonstruować wydajny algorytm faktoryzacji.

6.2.2 Wersje pełne

Schematy podane w Rozdziale 6.2.1 nie są w pełni bezpieczne. Np. w przypadku schematu szyfrowania Ewa może zauważyć, że dana wiadomość została wysłana wielokrotnie (bo za każdym razem zobaczy ten sam szyfrogram), poza tym jeśli Ewa podejrzewa, że Alicja otrzyma od Boba konkretną wiadomość np. $m_0 = 13242$, to może to zweryfikować (obliczając samemu szyfrogram dla m_0 i porównując z szyfrogramami przesyłanymi przez Boba). (Problemy podobnej natury pojawiają się w przypadku schematu podpisywania.) Poza tym pokazano ([3], patrz też [2]), że na podstawie $E_{n,e}^{\text{RSA}}(x)$ można obliczyć wartość symbolu Jacobiego x .

Aby zaradzić tym problemom stosuje się dwa podejścia: praktyczne i teoretyczne. W podejściu praktycznym używa się technik randomizacji i haszowania. Pomijamy tu szczegóły (patrz np. [2]).

W podejściu teoretycznym stosuje się generyczne konstrukcje, które przekształcają dowolną funkcję jednokierunkową z zapadką (ang.: *trapdoor one-way function*) w asymetryczny schemat szyfrowania. Nieformalnie mówiąc: funkcja jednokierunkowa z

zapadką (lub po prostu: *funkcja z zapadką*), to taka funkcja jednokierunkowa którą łatwo odwrócić jeśli zna się dodatkową informację: *zapadkę* (ang.: *trapdoor*). Szczegółowa definicja znajduje się w [2]. Przykładem funkcji jednokierunkowej z zapadką jest RSA zdefiniowane jako

$$\text{RSA}(n, e, x) = x^e \pmod n.$$

Zapadką jest tu d , lub faktoryzacja n (gdzie n, e i d są zdefiniowane jak powyżej).

Mamy następujące twierdzenia:

Fakt 15 *Wydajne wielomianowo bezpieczne asymetryczne schematy szyfrowania istnieją wtedy i tylko wtedy gdy istnieją funkcje jednokierunkowe z zapadką.*

Fakt 16 *Wydajne wielomianowo bezpieczne schematy podpisu istnieją wtedy i tylko wtedy gdy istnieją funkcje jednokierunkowe.*

7 Inne przydatne fakty z teorii liczb

7.1 Generatory i grupy cykliczne

Niech G będzie dowolną multiplikatywną grupą skończoną. *Rzędem elementu* $a \in G$ nazywamy najmniejszą liczbę naturalną i taka, że $a^i = 1$. Jeśli rząd a jest równy $|G|$, to element a jest *generatorem grupy* G . Grupę posiadającą generator nazywamy *cykliczną*.

Fakt 17 *Grupa Z_n^* jest cykliczna wtedy i tylko wtedy gdy $n = 2, 4, p^k, 2p^k$ (gdzie p jest nieparzystą liczbą pierwszą, a k jest liczbą naturalną większą od 0).*

7.2 Logarytm dyskretny

Jeśli a, b są elementami jakiejś grupy multiplikatywnej G , to najmniejszą liczbę naturalną x taką, że

$$a^x = b$$

nazywamy *logarytmem dyskretnym* b o podstawie a . Jeśli a jest generatorem grupy G , to wartość ta jest określona dla dowolnego $b \in G$. Jeśli p jest liczbą pierwszą, to problem obliczania logarytmu dyskretnego w grupie Z_p jest uznawany za trudny. Dokładniej mówiąc, funkcja

$$\text{EXP}(p, g, x) := (p, g, g^x \pmod p)$$

jest kandydatem na funkcję jednokierunkową.

Na trudności obliczania logarytmu dyskretnego oparte jest bezpieczeństwo *kryptosystemu El Gamala*.

7.3 Reszty kwadratowe

Liczba $a \in Z_n^*$ jest *resztą kwadratową modulo n* jeśli $a = b^2 \pmod n$ dla pewnego $b \in Z_n^*$. Zbiór reszt kwadratowych oznaczmy przez QR_n .

Fakt 18 Dla dowolnej liczby $a \in QR_p$ (dla p - nieparzystej pierwszej). Istnieją dokładnie dwie różne liczby x, y , takie, że $x^2 = y^2 = a$.

Dowód: Niech $x^2 = a$. Wówczas $y = -x$. Z pewnością $x \neq -x$ (bo wtedy mielibyśmy $2x = 0 \pmod p$, czyli x byłoby podzielne przez p). Załóżmy, że mamy $z \notin \{x, -x\}$, takie, że $z^2 = a$. Wówczas $z^2 - x^2 = 0$. A zatem

$$(z - x)(z + x) = 0 \pmod p$$

Wówczas któreś z $(z - x)$ i $(z + x)$ (obydwa są niezerowe) musiałyby być podzielne przez p . Sprzeczność. \square

Zatem liczb które są QR jest dokładnie połowa ze wszystkich elementów w Z_p^* . Konkretnie łatwo można pokazać, że są to wszystkie liczby postaci

$$g^{2m}$$

gdzie g jest generatorem. Z Chińskiego Twierdzenia o Resztach wiemy, że dla dowolnego $a \in QR_{pq}$ istnieją 4 rozwiązania równania

$$x^2 = a \pmod{pq}$$

Rozwiązania te można je wydajnie znaleźć, jeśli tylko potrafimy rozwiązywać podobne równania w liczbach pierwszych.

Lemat 19 Niech p będzie liczbą pierwszą i niech $a \in QR_p$. Istnieje WAP A taki, że $A(p, a) = x$, gdzie $x^2 = a \pmod p$.

Lemat 20 Obliczanie pierwiastków modulo $n \in H_k$ jest tak samo trudne jak faktoryzacja.

Dowód: W jedną stronę właśnie pokazaliśmy. W drugą: niech I będzie algorytmem który dla danego wejścia $n \in H_k$ i $a \in QR_n$ zwraca y , tżę $a = y^2 \pmod n$. Konstruujemy algorytm który na wejściu n zwraca nietrywialny dzielnik n .

1. Załóżmy $x \in Z_n^*$.
2. Niech $y = I(n, x^2 \pmod n)$
3. Sprawdź, czy $y \in \{x, -x\}$. Jeśli nie, to $\gcd(x - y, n)$ jest nietrywialnym dzielnikiem n . Wpp idź do 1.

• **Poprawność** $x^2 = y^2 \pmod n \implies (x + y)(x - y) = 0 \pmod n$. Stąd

$$n \mid (x + y)(x - y)$$

Zauważmy, że $n \nmid (x - y)$ bo $x \neq y \pmod n$ i $n \nmid (x + y)$ bo $x \neq -y \pmod n$.

Dlatego $\gcd(x - y, n)$ jest nietrywialnym dzielnikiem n .

- **Terminacja** Ponieważ liczba x^2 ma 4 pierwiastki, to szansa, że $y \in \{x, -x\}$ wynosi $\frac{1}{2}$. Stąd prawdopodobieństwo, że wrócimy do p. 1 wynosi $\frac{1}{2}$. Czyli szansa, że będziemy się pętlić k razy wynosi co najwyżej 2^{-k} .

□

Na trudności obliczania pierwiastków w Z_{pq} oparte jest bezpieczeństwo *kryptosystemu Rabina*.

Literatura

- [1] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.
- [2] S. Goldwasser and M. Bellare. Lecture notes in cryptography. dostępne pod adresem <http://www.cs.ucsd.edu/users/mihir/papers/gb.html>.
- [3] R. Lipton. How to cheat in mental poker. In *Proc. AMS Short Course in Cryptography*, 1981.
- [4] N. Saxena M. Agrawal, N. Kayal. PRIMES is in P. dostępne na stronie <http://www.cse.iitk.ac.in/users/manindra/>.
- [5] A. J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of applied cryptography*. CRC Press, 1997.
- [6] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. *Communications of the Association for Computing Machinery*, 21(2):120–126, February 1978.
- [7] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657–715, 1949.