

## Wykład 1. Wprowadzenie do kryptografii (część 1)

Stefan Dziembowski stefan-mpe@dziembowski.net

*Streszczenie.* Rozpoczynamy nieformalne wprowadzenia do teoretycznej kryptografii. Na przykładzie pojęcia szyfrowania wyjaśniamy podstawowe pojęcia używane w tej dziedzinie.

## 1 Szyfry

W tym rozdziale zajmujemy się definicją pojęcia *szyfru*, czyli (ogólnie mówiąc) metody zapewniania tajności komunikacji. Nieformalna definicja *tajności* szyfru stanowi dla nas punkt wyjścia do prób sformalizowania tego pojęcia.

### 1.1 Szyfry — wstęp do definicji

Klasycznym zadaniem kryptografii jest stworzenie metod *bezpiecznej* komunikacji. Na dzisiejszym wykładzie zastanawiać się będziemy nad precyzyjnym zdefiniowaniem tego pojęcia. Dla ustalenia uwagi załóżmy, że mamy do czynienia z dwiema osobami *Alicją* i *Bobem*. Osoby te mogą komunikować się ze sobą jedynie za pomocą łącza, które może być *podsluchiwane* przez ich przeciwnika *Ewę*. Przyjmijmy na razie, że nie jest możliwa ingerencja Ewy w treść przekazanej wiadomości (w szczególności Ewa nie może podszyć się pod Alicję ani pod Boba). Przykładem takiego łącza jest linia telefoniczna (gdy Alicja i Bob potrafią poznać się po głosie). W tym rozdziale pokażemy jak Alicja i Bob mogą wykorzystać powyższe łącze w celu przekazywania sobie wiadomości w taki sposób, by Ewa nie miała na jej temat żadnej informacji<sup>1</sup> (poza, ewentualnie, jej długością). Innymi słowy: by zapewniona była *tajność* przekazywanych wiadomości. Aby umożliwić Alicji i Bobowi sekretną komunikację wyposażamy ich w *szyfr*. Nieco uproszczając sprawę zdefiniujemy *szyfr* jako piątkę  $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{D})$ , gdzie:

- $\mathcal{K}$  jest *zbiorem kluczy*,
- $\mathcal{M}$  jest *zbiorem wiadomości*,
- $\mathcal{C}$  jest *zbiorem szyfrogramów*,
- $\mathcal{E} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$  jest *funkcją szyfrującą*
- $\mathcal{D} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$  jest *funkcją deszyfrującą*

Zakładamy, że Alicja i Bob są w posiadaniu (znanego tylko im) klucza  $k$  wylosowanego ze zbioru  $\mathcal{K}$ . Aby *zaszyfrować* wiadomość  $m \in \mathcal{M}$  Alicja oblicza *kryptogram*  $c = \mathcal{E}(k, m)$ . Następnie Alicja wysyła  $c$  do Boba (po łączu potencjalnie podsłuchiwanym przez Ewę). W celu *odszyfrowania* wiadomości Bob oblicza  $m' = \mathcal{D}(k, c)$ . Oczywiściewym wymaganiem jest żeby  $m = m'$ , czyli:

<sup>1</sup>Na tym etapie nie nadajemy temu pojęciu żadnego formalnego znaczenia.

**Postulat 1** Dla dowolnych  $k \in \mathcal{K}$  i  $m \in \mathcal{M}$  mamy

$$\mathcal{D}(k, \mathcal{E}(k, m)) = m$$

Ponieważ interesują nas jedynie *praktyczne* metody szyfrowania, mamy ponadto następujące wymaganie.

**Postulat 2** Funkcje  $\mathcal{E}$  i  $\mathcal{D}$  są wydajnie obliczalne.

Niezwykle ważne w kryptografii jest dokładne podanie które elementy szyfru uważamy za tajne. Najzdrowszym podejściem jest uznanie, że wszystko co nie jest wyraźnie określone jako tajne, jest znane przeciwnikowi. W naszym przypadku zaznaczyliśmy jedynie, że tajny jest klucz  $k$ . W związku z tym musimy założyć co następuje:

**Postulat 3** Przeciwnik zna funkcje  $\mathcal{E}$  i  $\mathcal{D}$ .

Zauważmy, że im bardziej pesymistyczne założenia przyjmiemy przy konstrukcji szyfru, tym mocniejszy szyfr otrzymamy. W szczególności, jeśli oparliśmy bezpieczeństwo szyfru na utajnieniu  $\mathcal{E}$  lub  $\mathcal{D}$ , to ujawnienie tych funkcji (np. za pomocą metod tzw. *reverse-engineering*) spowodowałoby załamanie bezpieczeństwa całego systemu. Wymieńmy wreszcie najważniejszą (a zarazem natrudniejszą w analizie) cechę, którą powinien mieć każdy szyfr:

**Postulat 4** Jeśli Ewa nie zna klucza  $k$ , to wiadomość  $m$  pozostaje tajna dla Ewy, nawet jeśli zna ona odpowiadający jej szyfrogram  $c = \mathcal{E}(k, m)$  (oraz funkcje  $\mathcal{E}$  i  $\mathcal{D}$ ).

Co jednak oznacza *tajność* w tym kontekście? Odpowiedź na to pytanie jest złożona. Zanim jej udzielimy przedstawimy prosty przykład szyfru (w celu przybliżenia odpowiednich intuicji).

## 1.2 Szyfr Vernama

Niech  $\oplus : \{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$  oznacza operację dodawania modulo 2 (to znaczy niech  $a \oplus b := a + b \pmod{2}$ ). Niech  $n$  będzie pewną ustaloną liczbą naturalną. Rozszerzmy działanie operacji  $\oplus$  na wektory  $\{0, 1\}^n$ , w sposób następujący:

$$(a_1, \dots, a_n) \oplus (b_1, \dots, b_n) := (a_1 \oplus b_1, \dots, a_n \oplus b_n).$$

Oczywiście zbiór  $\{0, 1\}^n$  z operacją  $\oplus$  jest grupą abelową, której elementem neutralnym jest  $(0, \dots, 0)$ . Ponadto, dla dowolnego  $x \in \{0, 1\}^n$  mamy  $x \oplus x = (0, \dots, 0)$ . Szyfr *Vernama* [Ver26] zdefiniowany jest następująco:

- $\mathcal{M} := \{0, 1\}^n$ ,  $\mathcal{K} := \{0, 1\}^n$  i  $\mathcal{C} := \{0, 1\}^n$ ,
- funkcja szyfrująca dana jest wzorem  $\mathcal{E}_v(k, m) = k \oplus m$ ,
- funkcja deszyfrująca jest identyczna z szyfrującą:  $\mathcal{D}_v(k, c) = k \oplus c$ .
- Dodatkowo zakładamy, że dany klucz może być użyty tylko raz (stąd angielska nazwa szyfru Vernama: *One-Time Pad*).

**Lemat 1** Niech  $K$  będzie zmienną losową z rozkładem jednostajnym na zbiorze  $\mathcal{K}$ . Wówczas dla dowolnych dwóch wiadomości  $m, m' \in \mathcal{M}$  zmienne losowe  $\mathcal{E}_v(K, m)$  i  $\mathcal{E}_v(K, m')$  mają identyczny rozkład (co zapisujemy:  $\mathcal{E}_v(K, m) \stackrel{d}{=} \mathcal{E}_v(K, m')$ )

**Dowód:** Dla dowolnej wiadomości  $m \in \mathcal{M}$  i szyfrogramu  $c \in \mathcal{C}$  mamy

$$\begin{aligned} P[\mathcal{E}_v(K, m) = c] &= P[K \oplus m = c] \\ &= P[K = c \oplus m] \\ &= 2^{-n}. \end{aligned}$$

Zatem, niezależnie od wyboru  $m$ , zmienna  $\mathcal{E}_v(K, m)$  ma zawsze rozkład jednostajny na zbiorze  $\{0, 1\}^n$ .  $\square$

Powyższy lemat oznacza, że informacja uzyskana przez Ewę jest całkowicie niezależna (w sensie rachunku prawdopodobieństwa) od wartości zaszyfrowanej wiadomości. Intuicyjnie, powinno być to gwarancją całkowitej tajności i rzeczywiście warunek ten przyjmuje się za definicję *idealnej tajności*.

**Definicja 2** Szyfr  $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{D})$  nazywamy idealnie tajnym jeśli dla dowolnych  $m, m' \in \mathcal{M}$  mamy:

$$\mathcal{E}(K, m) \stackrel{d}{=} \mathcal{E}(K, m')$$

(gdzie  $K$  jest zmienną losową o rozkładzie jednostajnym na  $\mathcal{K}$ ).

(„ $\stackrel{d}{=}$ ” oznacza równość rozkładów). Z Lematu 1 wynika zatem, że szyfr Vernama jest idealnie tajny. W tym momencie wydawać by się mogło, że szyfr Vernama rozwiązuje kwestię poszukiwania dobrego szyfru. Jest on bowiem jednocześnie prosty, wydajny i zapewnia idealną tajność. Podstawową przyczyną dla której szyfr Vernama jest rzadko stosowany w praktyce jest fakt, że długość klucza musi być równa długości wiadomości oraz, że dany klucz może być użyty co najwyżej raz.

Aby zilustrować ten problem rozważmy sytuację w której do zaszyfrowania dwóch wiadomości  $m$  i  $m'$  użyto tego samego klucza  $k$ . Ewa dysponująca szyfrogramami  $c = m \oplus k$  oraz  $c' = m' \oplus k$  może teraz obliczyć  $c \oplus c' = (m \oplus k) \oplus (m' \oplus k) = m \oplus m'$ . Jeśli Ewa domyśla się początku wiadomości  $m$  (np. wie, że jest to binarna reprezentacja napisu „Szanowny Panie!”), to z łatwością obliczy początek  $m'$ . Trudno uznać więc tę metodę za bezpieczną.<sup>2</sup> Podobne problemy napotkamy jeśli chcielibyśmy w naiwny sposób użyć krótszego klucza w szyfrze Vernama (np. używając niektórych bitów klucza wielokrotnie).

Z Teorii Informacji Shannona [Sha49] wynika, że problem ten nie jest przypadkowy. Zachodzi bowiem następujący fakt.

**Twierdzenie 3 ([Sha49])** Jeśli szyfr  $(\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{D})$  jest idealnie tajny, to  $|\mathcal{K}| \geq |\mathcal{M}|$ .

W następnym rozdziale pokażemy jak „obejść” to twierdzenie przez rezygnację z wymagania by tajność była idealna.

### 1.3 Szyfry stosowane w praktyce

W zastosowaniach używa się szyfrów które nie spełniają definicji idealnej tajności, mają jednak krótki klucz (który może być użyty wielokrotnie), są wydajne a przy tym (jak się wydaje) gwarantują wystarczającą tajność z punktu widzenia potrzeb praktycznych. Szyframi takimi są szyfry blokowe (np.: DES, AES, IDEA) w odpowiednim trybie użycia (np. CBC) i strumieniowe (np. RC4). Ich szczegółowe omówienie wykracza poza tematykę wykładu (zainteresowani znajdą więcej informacji np. w [MvOV97]).

Bezpieczeństwo tych szyfrów jest problemem otwartym. Aby jednak wogóle taki problem formalnie postawić potrzebna jest definicja tajności która jest słabsza niż tajność idealna,

---

<sup>2</sup>venona

dostatecznie jednak mocna, by była akceptowalna z praktycznego punktu widzenia. Jak się okazuje właściwego aparatu pojęciowego dostarcza dziedzina zwana *teorią złożoności obliczeniowej*. Do praktycznych zasotosowań wystarczają bowiem szyfry, które są tajne ze względu na przeciwnika o ograniczonej mocy obliczeniowej.

## 1.4 Poszukiwanie słabszej definicji tajności

Zastanówmy się teraz jak taka (słabsza) definicja tajności mogłaby wyglądać. W tym rozdziale przedstawimy kilka narzucających się propozycji takiej definicji (i wskażemy ich wady). (We wszystkich propozycjach zakładamy, że  $K$  jest zmienną losową o jednostajnym rozkładzie na  $\mathcal{K}$ ). Pierwszym pomysłem, który przychodzi do głowy jest zdefiniowanie bezpieczeństwa szyfru przez trudność obliczenia wiadomości na podstawie szyfrogramu:

**Propozycja 1** *Nie istnieje wydajny algorytm  $\mathcal{A}$  pobierający na wejściu  $\mathcal{E}(K, m)$  i zwracający  $m$ .*

(Nie precyzujemy na razie pojęcia „wydajny”). Jak się okazuje powyższa propozycja jest zbyt słaba (to znaczy nie gwarantuje bezpieczeństwa w praktyce). Jako przykład rozważmy szyfr Vernama w którym do szyfrowania użyto tego samego klucza dwukrotnie (nazwijmy go  $S_{v2}$ ):  $\mathcal{M}_{v2} := \{0, 1\}^{2n}$ ,  $\mathcal{K}_{v2} := \{0, 1\}^n$  i  $\mathcal{C}_{v2} := \{0, 1\}^{2n}$ ,  $\mathcal{E}_{v2}(k, m) = (k \cdot k) \oplus m$ ,  $\mathcal{D}_{v2}(k, c) = (k \cdot k) \oplus c$  (gdzie „ $\cdot$ ” oznacza konkatencję). Szyfr ten nie gwarantuje tajności z dokładnie tych samych powodów dla których w szyfrze Vernama nie należy używać tego samego klucza dwukrotnie (patrz Rozdział 1.2). Z drugiej strony dla dowolnego szyfrogramu  $c$  istnieje  $2^n$  potencjalnych wiadomości  $m$  dla których istnieje klucz  $k$  taki, że  $\mathcal{E}_{v2}(k, m) = c$ . Rzeczywiście, jeśli  $c$  przedstawimy jako konkatencję  $c = c_1 \cdot c_2$  (gdzie  $|c_1| = |c_2| = n$ ), to potencjalne wiadomości tworzą zbiór

$$\mathcal{L}_c := \{m \cdot (m \oplus c_0 \oplus c_1) : m \in \{0, 1\}^n\}.$$

Szyfr jest więc tajny w sensie Propozycji 1. Z podobnych przyczyn nie ma sensu definiowanie bezpieczeństwa przez trudność obliczenia poszczególnych bitów zaszyfrowanej wiadomości. Zastanówmy się jak wzmocnić definicję z Propozycji 1 w taki sposób, by szyfr  $S_{v2}$  nie był według niej tajny. Zauważmy, że choć nie istnieje metoda obliczania wiadomości  $m$  na podstawie szyfrogramu  $c$ , to potrafimy znajomość  $c$  „pomaga” w zgadnięciu wiadomości  $m$ . Aby w sposób ścisły mówić, o szansach zgadnięcia  $m$  musimy przyjąć jakiś rozkład prawdopodobieństwa z jakim  $m$  zostało wybrane. Dla przykładu załóżmy, że  $m \in \mathcal{M}$  zostało wybrane z prawdopodobieństwem jednostajnym. Wówczas algorytm który na wejściu  $c$  zwraca dowolne  $m \in \mathcal{L}_c$  ma szanse trafienia  $2^{-n}$ . Oczywiście, bez znajomości  $c$  szanse te są znacznie mniejsze ( $2^{-2n}$ ). Podążając tą obiecującą drogą zaproponujemy nową definicję tajności:

**Propozycja 2** *Niech  $M$  będzie dowolną zmienną losową przyjmującą wartości z  $\mathcal{M}$  (niekoniecznie z rozkładem jednostajnym). Wówczas dowolny wydajny algorytm  $\mathcal{A}$  pobierający na wejściu  $c = \mathcal{E}(K, M)$  zwraca  $m$  z prawdopodobieństwem nie większym niż  $p_{\max}$ , gdzie*

$$p_{\max} = \max_{m \in \mathcal{M}} P[M = m].$$

Jak się okazuje tak sformułowana definicja tajności jest równoważna tajności idealnej (ćwiczenie). Jest tak dlatego, że wymaganie by prawdopodobieństwo zgadnięcia nie było większe niż  $p_{\max}$  jest zbyt restrykcyjne. Spróbujmy więc zastąpić tędefinicję nieco bardziej liberalną:

**Propozycja 3** *Niech  $M$  będzie dowolną zmienną losową przyjmującą wartości z  $\mathcal{M}$ . Wówczas dowolny wydajny algorytm  $\mathcal{A}$  pobierający na wejściu  $c = \mathcal{E}(K, m)$  zwraca  $m$  z prawdopodobieństwem niewiele większym niż  $p_{\max}$ .*

To z kolei rodzi pytanie: co oznacza „niewiele”. Problem polega na tym, że jeśli  $M$  ma np. rozkład jednostajny na  $\{0, 1\}^n$  (dla dużego  $n$ ), to w przypadku szyfru  $S_{v2}$  szanse sukcesu dowolnego algorytmu  $\mathcal{A}$  są zawsze niewiele większe od zera (a więc i od  $p_{\max}$ ), bo są równe  $2^{-n}$ .

W ten sposób powstaje pomysł, by ograniczyć się do zmiennych losowych  $M$  należących do jakiejś określonej klasy, np. do takich zmiennych które przyjmują tylko małą liczbę wartości. Jak się okazuje wystarczy ograniczyć się do zmiennych które przyjmują dwie wartości z prawdopodobieństwem jednostajnym.

**Propozycja 4** *Niech  $M$  będzie zmienną losową o rozkładzie jednostajnym na dowolnym dwuelementowym zbiorze  $\{m_0, m_1\} \subseteq \mathcal{M}$ . Wówczas dowolny wydajny algorytm  $\mathcal{A}$  pobierający na wejściu  $c = \mathcal{E}(K, M)$  zwraca  $m$  z prawdopodobieństwem niewiele większym niż 0.5.*

Jest to właściwy trop. Definicja oparta na tej propozycji gwarantuje bowiem bezpieczeństwo w praktyce (przynajmniej według obecnie panujących przekonań) a ponadto istnieją praktycznie używane szyfry które (prawdopodobnie) spełniają tę deficytę. W następnym rozdziale postaramy się powyższą propozycję ukonkretnić. Zanim to zrobimy zauważmy, że na powyższą definicję można patrzeć w kategoriach istnienia strategii wygrywającej (dla Alicji) w następującej grze:

1. Ewa wybiera parę  $m_0, m_1$  i wysyła ją do Alicji.
2. Alicja losuje klucz  $k$  i bit  $b \in \{0, 1\}$  i odsyła  $\mathcal{E}(k, m_b)$  do Alicji.
3. Ewa ma zgadnąć  $b$ .

Szyfr jest bezpieczny jeśli (ograniczona obliczeniowo) Ewa ma szanse niewiele większe od 0.5.

## 1.5 Tajność obliczeniowa

Po pierwsze musimy ustalić co oznacza pojęcie *wydajnego algorytmu*. Istnieje szereg różnych modeli obliczeń poczynając od bardziej bliskich rzeczywistości (np. model RAM) a kończąc na bardziej teoretycznych (maszyna Turinga, obwody logiczne). Dla ustalenia uwagi przyjmijmy model oparty na maszynach Turinga. Jeśli zostanie ustalony konkretny wariant definicji maszyny Turinga wówczas możemy Propozycję 4 przerobić na formalną definicję matematyczną:

**Definicja 4** *Szyfr  $S = (\mathcal{K}, \mathcal{M}, \mathcal{C}, \mathcal{E}, \mathcal{D})$  jest  $(\epsilon, t)$ -TM-tajny jeśli dla dowolnej pary wiadomości  $m_0, m_1 \in \mathcal{M}$  i dla maszyny Turinga  $\mathcal{A}$  pobierającej na wejściu  $m \in \mathcal{M}$  i  $c \in \mathcal{C}$  i działającej w  $t$  krokach:*

$$P[\mathcal{A}(\mathcal{E}(M, K)) = M] \leq \frac{1}{2} + \epsilon,$$

gdzie prawdopodobieństwo jest liczone po  $M \xleftarrow{r} \{m_0, m_1\}$  i  $K \xleftarrow{r} \mathcal{K}$ .

Zauważmy, że w definicji tej musieliśmy jawnie podać wartości  $\epsilon$  i  $t$ . Prowadzi to do następującej obserwacji: zamiast mówić o istnieniu bezpieczeństwa bądź o jego braku lepiej mówić o *stopniu bezpieczeństwa*. (Uwaga ta prawdziwa jest zresztą także w odniesieniu do szeregu innych aspektów bezpieczeństwa systemów komputerowych.) Dla pewnych zastosowań wystarczy np.  $(2^{-50}, 2^{50})$ -TM-tajność, a dla innych może to być za mało. Także: to co teraz jest uważane za bezpieczne, nie musi być takie w przyszłości (kto wie jakie będą moce obliczeniowe za 100 lat).

Z punktu widzenia analizy teoretycznej podejście takie jak w Definicji 4 ma jednak szereg wad. Po pierwsze wiąże nas z konkretnym modelem obliczeń, po drugie jest nieporęczne w stosowaniu ze względu na symbole  $\epsilon$  i  $t$  (jak się okazuje w wielu dowodach bezpieczeństwa wzory na  $\epsilon$  i  $t$  przybierają koszarne formy). Z tych względów zwykle upraszcza się pojęcie tajności. Wypróbowaną metodą jest *podejście asymptotyczne*. W podejściu tym wprowadzamy z reguły zmienną  $i \in \mathbf{N}$  zwaną *parametrem bezpieczeństwa*, która jest argumentem dla funkcji szyfrującej i odszyfrowującej. Dzięki temu, zamiast pojedynczego szyfru  $S$  otrzymujemy nieskończoną rodzinę szyfrów  $\mathcal{S}$  (dla każdego  $i$  osobny szyfr). W związku z tym  $\epsilon$  i  $t$  w definicji tajności stają się funkcjami zmiennej  $i$  (dla odróżnienia oznaczmy je przez  $E$  i  $T$ , odpowiednio). Jeśli teraz skoncentrujemy się wyłącznie na zachowaniu funkcji  $\mathcal{E}$  i  $T$  w nieskończoności, to pozwoli nam to na abstrakcję od szeregu problemów technicznych (takich jak precyzyjne zdefiniowanie modelu obliczeń).

Jak się okazuje najwygodniejszym do tego celu pojęciem jest pojęcie wielomianu. Konkretnie: przez wydajne obliczenie będziemy rozumieć takie, które jest wykonywane przez maszynę Turinga w wielomianowej (ze względu na długość wejścia) liczbie kroków (to znaczy: istnieje taki wielomian  $p$ , że dla wejścia o rozmiarze  $x$  maszyna wykonuje maksymalnie  $p(x)$  kroków). Podkreślimy zalety tego podejścia: nie ma znaczenia, jaką definicję maszyny Turinga przyjęliśmy (ile jest taśm, jaki jest alfabet, etc.). Nie ma zresztą wogóle znaczenia, że zdecydowaliśmy się na maszynę Turinga. Równie dobrze mogłaby to być np. maszyna RAM. Jest tak dlatego, że wszystkie modele te są sobie równoważne w sensie redukcji w czasie wielomianowym (tzn. maszynę Turinga można zasymulować na maszynie RAM i odwrotnie). Ponadto podejście to pozwoli nam w szeregu dowodów pozbyć się denerwujących komplikacji we wzorach. Dzieje się tak dlatego, że klasa wielomianów jest zamknięta na operacje takie jak: dodawanie, mnożenie oraz składanie.

Przez *wielomianowy algorytm probabilistyczny (WAP)* będziemy rozumieć algorytm działający w czasie wielomianowym (ze względu na długość wejścia), który dodatkowo ma prawo rzucać monetą (formalnie modelujemy ten obiekt jako maszynę Turinga wyposażoną w taśmę na której zapisać nieskończony ciąg losowo wybranych zer i jedynek).

Powiemy, że funkcja  $E : \mathbf{N} \rightarrow \mathbf{R}$  jest *zaniedbywalna* jeśli dla dowolnego wielomianu  $p$  istnieje  $N$  takie, że dla każdego  $i > N$  mamy  $|E(i)| \leq \frac{1}{p(i)}$ . Inaczej mówiąc: funkcja jest zaniedbywalna jeśli jej wartość maleje do zera szybciej niż odwrotność dowolnego wielomianu. Pojęcie to okazuje się wygodne dlatego, że funkcja zaniedbywalna pomnożona przez dowolny wielomian jest cały czas zaniedbywalna. Ponadto że klasa funkcji zaniedbywalnych jest zamknięta ze względu na dodawanie, mnożenie i składanie. Przykładem funkcji zaniedbywalnej jest funkcja dana wzorem  $E(i) = 2^{-i}$ .

**Definicja 5** *Szyfrem nazywamy trójkę  $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ , przy czym*

- $\mathcal{K}$  jest wielomianowym probabilistycznym algorytmem generacji klucza biorącym jako argument parametr bezpieczeństwa  $1^i$  i zwracającym klucz (niech  $\text{Keys}$  oznacza zbiór wartości które  $\mathcal{K}$  zwraca z niezerowym prawdopodobieństwem dla jakiegoś wejścia),
- $\mathcal{E}$  jest wielomianowym probabilistycznym algorytmem szyfrującym pobierającym na wejściu parę  $(k, m)$  (gdzie  $k \in \text{Keys}$  i  $m \in \{0, 1\}^*$ ) i zwracającym szyfrogram  $c \in \{0, 1\}^*$ ,
- $\mathcal{D}$  jest wielomianowym probabilistycznym algorytmem deszyfrującym pobierającym na wejściu parę  $(k, c)$  (gdzie  $k \in \text{Keys}$  i  $c \in \{0, 1\}^*$ ) i zwracającym  $m \in \mathcal{M}$ .

Przy czym z prawdopodobieństwem 1 zachodzi  $\mathcal{D}(k, \mathcal{E}(k, m)) = m$ .

Uwagi:

1. Piszemy  $1^i$  zamiast  $i$  aby wymóc, że algorytm działa w czasie wielomianowym *ze względu na  $i$*  (a nie na  $\log(i)$ ).
2. Zamiast definiować *zbiór* kluczy mamy teraz *algorytm generacji klucza*. Jak się okazuje takie podejście jest zarówno wygodniejsze w teoretycznych rozważaniach, jak i bliższe praktyki.
3. Nie ma konieczności podawania parametru bezpieczeństwa algorytmom  $\mathcal{E}$  i  $\mathcal{D}$ , gdyż możemy założyć, że wartość tego parametru daje się policzyć na podstawie klucza  $k$ .
4. Dla zwiększenia ogólności (i pozbycia się parametru  $n$ ) przyjmujemy, że zbiorem wiadomości jest zbiór  $\{0, 1\}^*$ . Nie ma też potrzeby definiowania zbioru kryptogramów.

**Definicja 6** *Szyfr  $S$  jest tajny obliczeniowo jeśli dla dowolnego WAP  $\mathcal{A}$  i dla dowolnych dwóch wiadomości  $m_0$  i  $m_1$  (tej samej długości) poniższa funkcja (zmiennej  $i$ ):*

$$P \left[ \mathcal{A}(1^i, \mathcal{E}(K, M)) = M \right] - 0.5$$

(gdzie prawdopodobieństwo jest liczone po  $K \stackrel{r}{\leftarrow} \mathcal{K}(1^i)$  i  $M \stackrel{r}{\leftarrow} \{m_0, m_1\}$ ) jest zaniedbywalna.

Należy podkreślić, że stosując powyższe podejście upraszczamy sobie pracę teoretyczną za cenę pewnego oddalenia się od praktyki.

## 1.6 Rozszerzamy definicję

Zauważmy, że definicje z poprzedniego rozdziału stosują się do sytuacji gdy dany klucz prywatny jest użyty co najwyżej raz. W praktyce używamy tego samego klucza wielokrotnie. Aby odzwierciedlić to w naszych definicjach musimy rozszerzyć je następująco. W Definicji 5 pozwalamy, by algorytm szyfujący  $\mathcal{E}$  posiadał *stan* (to znaczy wewnętrzną zmienną, której wartość przechowywana jest pomiędzy kolejnymi wywołaniami algorytmu). Jeśli np. używamy szyfru Vernama z bardzo długim kluczem (zużywanym „po kawałku” w kolejnych wywołaniach), to stanem może być licznik oznaczający indeks ostatniego użytego bitu klucza.

Po drugie, w Definicji 6 musimy wziąć pod uwagę fakt, że przeciwnik mógł uzyskać pewną informację (na temat klucza) na podstawie poprzednich kryptogramów. Rozważane są (między innymi) następujące rodzaje ataków (które przeciwnik ma prawo wykonać zanim przystąpi do gry w odróżnianie szyfrogramów dwóch wybranych przez siebie wiadomości).

1. Atak ze znanym kryptogramem — przeciwnik może poznać dowolną liczbę<sup>3</sup> kryptogramów losowych wiadomości. Tego typu atak jest zwykle łatwy do przeprowadzenia w praktyce.
2. Atak ze znaną wiadomością — przeciwnik może poznać dowolną liczbę par (wiadomość, szyfrogram), przy czym wiadomości są wybrane losowo zgodnie z rozkładem znanym przeciwnikowi). Tego typu atak (lub zbliżony do niego) bywa spotykany w praktyce, zwłaszcza, że często szyfrowane są wiadomości zapisane w publicznie znanych formatach.
3. Atak z wybraną wiadomością — przeciwnik może poznać dowolną liczbę kryptogramów wybranych przez siebie wiadomości. Wyróżnia się dwa rodzaje ataków:

---

<sup>3</sup>Oczywiście przeciwnik ograniczony czasem wielomianowym nie może poznać większej liczby niż wielomianowa

- (a) atak wsadowy — wiadomości muszą być wybrane z góry,
  - (b) atak adaptacyjny — wiadomości mogą być dobierane na podstawie poprzednich odpowiedzi.
4. Atak z wybranym kryptogramem — tak jak w Punkcie 3 z tym, że przeciwnik może otrzymać wiadomości odpowiadające wybranym przez niego kryptogramom.

Ataki z Punktów 4 i 3 mogą wydawać się mało realistyczne. W rzeczywistości mogą być one jednak łatwiejsze do przeprowadzenia niż się wydaje. Po pierwsze (ta uwaga dotyczy ataku z wybraną wiadomością) przeciwnik ma czasami wpływ na szyfrowaną wiadomość (np. bombardując jakieś miasto możemy spowodować by nazwa tego miasta zaczęła pojawiać się w depe szach wroga<sup>4</sup>). Po drugie, przeciwnik może uzyskać chwilowy dostęp do urządzenia szyfrującego. Np. pracy [Ble98] przedstawiono atak z wybranym kryptogramem (na standard PKCS #1), który opierał się wyłącznie na komunikatach o błędach wysyłanych przez system odszyfrowujący.

Matematyczne zdefiniowanie bezpieczeństwa względem powyższych ataków pozostawiamy jako ćwiczenie. Zauważmy tylko, że bezpieczeństwo względem każdego z nich (oprócz ataku ze znanym kryptogramem) implikuje, że algorytm szyfrujący musi być zrandomizowany lub posiadać stan, (by zapewnić, że dwukrotne zaszyfrowanie tej samej wiadomości daje różne szyfrogramy).

## 1.7 Dowody tajności

Podstawowym celem teoretycznej kryptografii jest udowodnienie bezpieczeństwa szyfrów stosowanych w praktyce. Niestety, nie jest znany dowód tajności jakiegokolwiek szyfru (poza szyframi tajnymi idealnie). Potrafimy natomiast dowieść bezpieczeństwa niektórych szyfrów przy przyjęciu założeń dotyczących trudności obliczeniowej niektórych problemów. Rozumujemy wtedy w ten sposób: jeśli dany szyfr  $S$  nie jest tajny to pewien problem uważany za trudny obliczeniowo (np. faktoryzacja iloczynów dużych liczb pierwszych) jest trudny. Więcej na ten temat na następnych wykładach.

## 2 Oznaczenia

„ $\stackrel{r}{\leftarrow}$ ” Jeśli  $\mathcal{X}$  jest zbiorem, to  $X \stackrel{r}{\leftarrow} \mathcal{X}$  oznacza, że zmienna losowa  $X$  ma rozkład jednostajny na  $\mathcal{X}$  (i jest niezależna od pozostałych zdefiniowanych zmiennych losowych).

Jeśli  $\mathcal{A}$  jest algorytmem probabilistycznym, to  $A \stackrel{r}{\leftarrow} \mathcal{A}(x)$  oznacza, że zmienna losowa  $A$  jest zdefiniowana jako wynik działania  $\mathcal{A}$  na argumentach  $x$ .

„ $\stackrel{d}{\equiv}$ ” Jeśli  $A$  i  $B$  są zmiennymi losowymi, to  $A \stackrel{d}{\equiv} B$  oznacza, że rozkład  $A$  jest identyczny z rozkładem  $B$ , to znaczy dla dowolnego  $x$  (które jest możliwą wartością przyjmowaną przez te zmienne) zachodzi

$$P[A = x] = P[B = x].$$

## Literatura

- [Ble98] Daniel Bleichenbacher. Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1. *Lecture Notes in Computer Science*, 1462:1–12, 1998.

---

<sup>4</sup>enigma



- [MvOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997. dostępne pod adresem <http://www.cacr.math.uwaterloo.ca/hac/>.
- [Sha49] C. E. Shannon. Communication theory of secrecy systems. *Bell Sys. Tech. J.*, 28:657–715, 1949.
- [Ver26] G. S. Vernam. Cipher printing telegraph systems for secret wire and radio telegraphic communications. *J. Am. Inst. Elec. Eng.*, 55:109–115, 1926.

*Data ostatniej modyfikacji: 7 października 2004.  
Wszelkie uwagi proszę zgłaszać na adres [stefan-mpe@dziembowski.net](mailto:stefan-mpe@dziembowski.net).  
Proszę nie rozpowszechniać tego dokumentu poza MIM UW bez mojej zgody.*