William Stallings

Wykorzystaj fascynujące możliwości kryptografii — *zapewnij bezpieczeństwo informacjom i sieciom komputerowym!*

KRYPTOGRAFIA *i* BEZPIECZEŃSTWO *sieci komputerowych*

Matematyka szyfrów i techniki kryptologii

Wydanie V

- Opanuj klasyczne techniki szyfrowania i wstęp do tech niki
- Poznaj skuteczne algorytmy ochrony integralności danych
- Stosuj kody uwierzytelniające komunikaty i podpisy cyfrowe

Helion

Kryptografia i bezpieczeństwo sieci komputerowych. Matematyka szyfrów i techniki kryptologii

Autor: William Stallings
Tłumaczenie: Andrzej Grażyński
ISBN: 978-83-246-2986-2
Tytuł oryginału: Cryptography and Network Security: Principles and Practice (5th Edition) , vol. 1
Format: 170×230, stron: 760

Wykorzystaj fascynujące możliwości kryptografii – zapewniaj bezpieczeństwo informacjom i sieciom komputerowym!

- Opanuj klasyczne techniki szyfrowania i wstęp do teorii liczb
- Poznaj skuteczne algorytmy ochrony integralności danych
- Stosuj kody uwierzytelniające komunikaty i podpisy cyfrowe

Wirusy, hakerzy, szpiegostwo gospodarcze, elektroniczne podsłuchy i kradzieże – era Internetu ma także swoją ciemną stronę, która stawia przed nami coraz większe wyzwania w zakresie bezpieczeństwa informacji. Dla większości przedsiębiorstw i organizacji kwestia ochrony dostępu do danych przechowywanych w systemach komputerowych i wymienianych między nimi, a także zachowania tajności wiadomości oraz skuteczne odpieranie ataków sieciowych, stała się zagadnieniem krytycznym, mogącym przesądzać o ich istnieniu. Bezpieczeństwo sieci ma także ogromne znaczenie także zwykłych użytkowników Internetu, często przetrzymujących na dyskach ważne, poufne dokumenty i dokonujących za pomocą Sieci rozmaitych finansowych transakcji.Na szczęście po ponad 20 latach od upowszechnienia się Internetu mamy już przetestowane w boju, dojrzałe technologie i narzędzia związane z bezpieczeństwem sieci komputerowych i kryptografią, które dają dziś naprawdę ogromne możliwości w tym zakresie. Jedyne czego Ci zatem potrzeba to uzbroić się w wiedzę jak je skutecznie wykorzystać.

Oto pierwszy z dwóch tomów kompletnego przewodnika po praktycznych zastosowaniach kryptografii i innych mechanizmów bezpieczeństwa w celu ochrony informacji i sieci. Ten adresowany zarówno do studentów, jak i zawodowców podręcznik podzielono na trzy naszpikowane wiedzą i ciekawymi przykładami części, wprowadzające kolejno w szyfry symetryczne, szyfry asymetryczne i kryptograficzne algorytmy ochrony integralności danych. Znajdziesz tu omówienia rozmaitych technologii związanych z bezpieczeństwem sieciowym, oraz poznasz metody ich implementacji i zastosowania. Przeczytasz m.in na temat trybów operacyjnych szyfrów blokowych, przyjrzysz się także standardowi AES i generowaniu liczb pseudolosowych. Otrzymasz obszerną, porównawczą prezentację algorytmów kryptograficznych i doskonały przewodnik po metodach uwierzytelniania i tematyce cyfrowego podpisu. Ponadto nauczysz się efektywnie wykorzystywać system Sage - wieloplatformowe, darmowe narzędzie implementujące użyteczny, elastyczny i łatwy do opanowania system obliczeń algebraicznych związanych z kryptografią. Znajdziesz także gotowe dla tego systemu przykłady, ilustrujące praktyczne zastosowania teorii liczb i algorytmów kryptograficznych.

Helion SA
ul. Kościuszki 1c
44-100 Gliwice
tel. 32 230 98 63
e-mail: helion@helion.pl

SPIS TREŚCI

ROZDZIAŁ 2

KLASYCZNE TECHNIKI SZYFROWANIA

Jestem za pan brat ze wszystkimi formami tajemnego pisma, sam przecież jestem autorem drobnej monografii na ten temat, w której analizuję 160 różnych szyfrów — rzekł Holmes.

— *The Adventure of the Dancing Men*, Arthur Conan Doyle

KLUCZOWE POJĘCIA

- Szyfrowanie symetryczne, zwane także szyfrowaniem konwencjonalnym, jest odmianą kryptosystemu, w którym zarówno szyfrowanie, jak i deszyfracja wykonywane są przy użyciu tego samego klucza.
- Szyfrowanie symetryczne transformuje tekst jawny na szyfrogram przy użyciu tajnego klucza i algorytmu szyfrowania. Stosując do szyfrogramu algorytm odwrotny z tym samym kluczem, otrzymujemy z powrotem tekst jawny.
- Dwa typy ataków, jakie intruzi mogą przypuścić na algorytm szyfrowania, to kryptoanaliza bazująca na właściwościach tego algorytmu oraz atak siłowy (brute-force) polegający na wypróbowywaniu wszystkich możliwych kluczy.
- Tradycyjne (przedkomputerowe) szyfrowanie opiera się na dwóch technikach: podstawieniowych i (lub) przestawieniowych. Techniki podstawieniowe dokonują odwzorowywania elementów tekstu jawnego (znaków lub bitów) na elementy szyfrogramu, techniki przestawieniowe opierają się na systematycznych zamianach pozycji elementów tekstu jawnego.
- Maszyny wirnikowe to wymyślny sprzęt epoki przedkomputerowej, implementujący techniki podstawieniowe.
- Steganografia to technika ukrywania tajnego komunikatu w obszerniejszym strumieniu danych w taki sposób, by osoby niepowołane nie mogły nawet zauważyć samego istnienia ukrytej informacji.

Szyfrowanie symetryczne, zwane również szyfrowaniem konwencjonalnym lub szyfrowaniem z pojedynczym kluczem, było jedyną metodą szyfrowania do czasu wynalezienia kryptografii z kluczami publicznymi w latach 70. ubiegłego stulecia, i pozostaje do dziś dominującą techniką szyfrowania. W pierwszej części książki omawiamy kilka szyfrów symetrycznych: ten rozdział rozpoczynamy od przedstawienia ogólnego modelu tego typu szyfrowania, co może okazać się pomocne w zrozumieniu kontekstu, w jakim stosowane są symetryczne algorytmy szyfrujące. Następnie omawiamy kilka popularnych algorytmów, stosowanych szeroko w czasach ery przedkomputerowej. Rozdział kończymy omówieniem kilku technik określanych wspólnym mianem steganografii. W rozdziałach 3. i 5. omówimy natomiast dwa najbardziej obecnie rozpowszechnione szyfry — DES i AES.

Na początek zdefiniujmy kilka podstawowych terminów. Oryginalny komunikat, podlegający szyfrowaniu, nazywamy **tekstem jawnym** (*plaintext*), a wynik jego zaszyfrowania — **szyfrogramem** (*ciphertext*). Proces konwertowania tekstu jaw-

nego na szyfrogram nazywamy **szyfrowaniem** lub **kryptażem** (*enciphering* lub *encryption*), zaś proces odwrotny, czyli odzyskiwanie tekstu jawnego na podstawie szyfrogramu — **deszyfracją** lub **dekryptażem (***deciphering* lub *decryption***)**. Ogół schematów składających się na szyfrowanie, zwanych (po prostu) **szyframi** lub **systemami kryptograficznymi** tworzy gałąź wiedzy zwaną **kryptografią**. Techniki wykorzystywane do uzyskania tekstu jawnego bez jakiejkolwiek wiedzy dotyczącej szczegółów szyfrowania określamy mianem **kryptoanalizy —** znanej także jako „łamanie kodu". Kryptografia i kryptoanaliza składają się na dziedzinę nauki zwaną **kryptologią**.

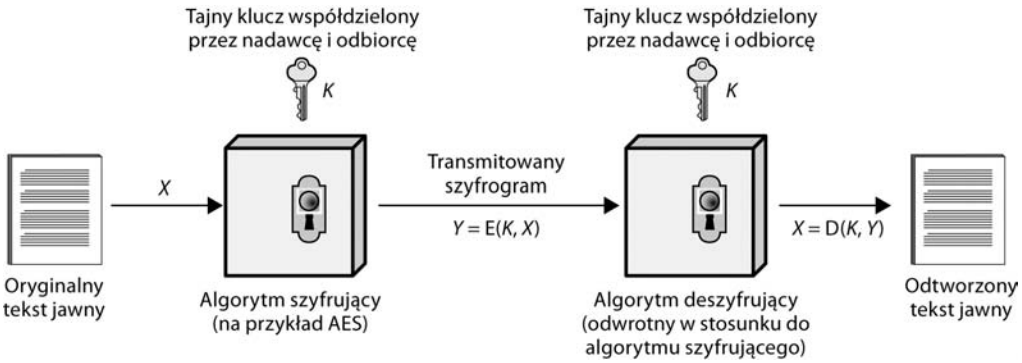
2.1. MODEL SZYFROWANIA SYMETRYCZNEGO

Schemat szyfrowania symetrycznego składa się z pięciu elementów, którymi są (patrz rysunek 2.1):

- **Tekst jawny** (*plaintext*) — oryginalny, czytelny komunikat (lub inne dane), stanowiący materiał wejściowy dla algorytmu szyfrującego.
- **Algorytm szyfrujący** (*encryption algorithm*) — algorytm dokonujący rozmaitych transformacji podstawieniowych i przestawieniowych na tekście jawnym.
- **Tajny klucz** (*secret key*) — parametr wejściowy określający szczegóły działania algorytmu szyfrującego, niezależny od samego algorytmu ani od teksu jawnego. Zastosowanie różnych kluczy do tego samego tekstu jawnego w tym samym czasie skutkuje różnymi wynikami — konkretne przestawienia i podstawienia wykonywane przez algorytm zależne są od konkretnego klucza.
- **Szyfrogram** (*ciphertext*) — zakodowany komunikat produkowany przez algorytm szyfrujący, zależny od tekstu jawnego i użytego klucza. Dla danego tekstu jawnego użycie dwóch różnych kluczy daje w efekcie różne szyfrogramy. Szyfrogram powinien mieć postać chaotycznego ciągu znaków, sprawiającego złudzenie losowego, co sprawi, że będzie on nieczytelny w sposób bezpośredni.
- **Algorytm deszyfrujący** (*decryption algorithm*) **—** algorytm odwrotny do algorytmu szyfrującego, odtwarzający tekst jawny na podstawie szyfrogramu i klucza, przy użyciu którego szyfrogram ten został utworzony.

Aby szyfrowanie konwencjonalne mogło zapewnić odpowiedni poziom bezpieczeństwa, konieczne jest spełnienie dwóch wymagań:

1. Algorytm szyfrujący musi być na tyle solidny, by znający go intruz, dysponujący dodatkowo zestawem szyfrogramów, nie był w stanie odtworzyć tekstu jawnego bez znajomości użytego klucza (kluczy). W praktyce wymaganie to formułowane jest w formie bardziej rygorystycznej: intruz nie



Rysunek 2.1. Uproszczony model szyfrowania symetrycznego

powinien mieć możliwości odtworzenia użytego klucza (kluczy) nawet wówczas, gdy dysponuje zestawem odpowiadających sobie par „tekst jawny – – szyfrogram".

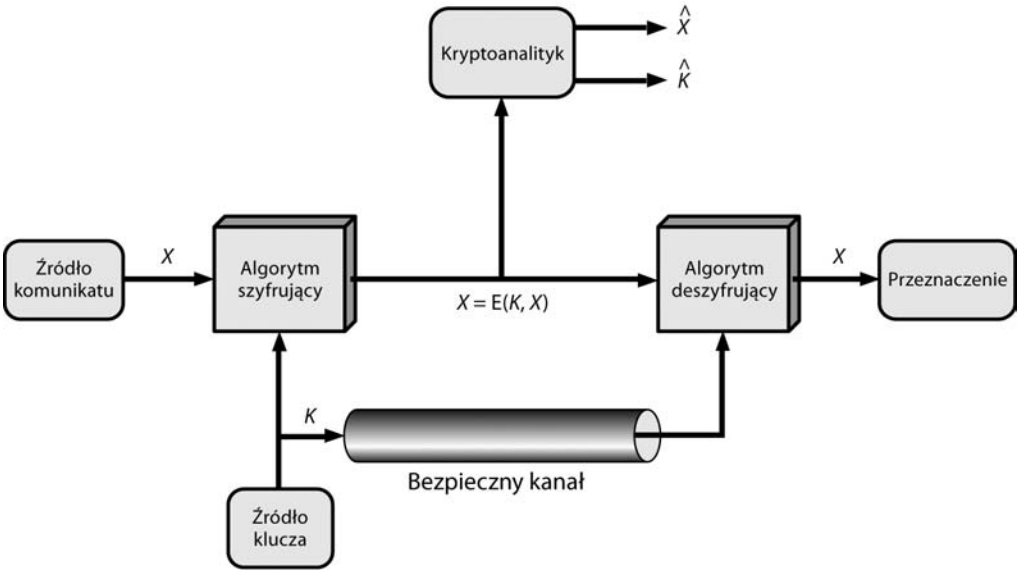
2. Nadawca i odbiorca muszą otrzymać kopie tajnego klucza w sposób bezpieczny i zachować je w tajemnicy. Gdy intruz pozna wspomniany klucz, szyfrowanie przy użyciu tego klucza stanie się bezcelowe.

Transformacja przekształcająca tekst jawny na szyfrogram zależna jest od dwóch elementów: algorytmu szyfrującego i klucza, a więc w celu zapewnienia jak największego bezpieczeństwa powinniśmy utrzymywać oba te elementy w tajemnicy. Mimo to ze względów praktycznych rezygnuje się z utajniania algorytmu szyfrującego, utrzymując w tajemnicy jedynie klucz. Ujawnienie używanego algorytmu szyfrującego umożliwia producentom sprzętu jego implementowanie w seryjnie produkowanych (a więc tanich) chipach, które znajdować mogą zastosowanie w szerokiej gamie produktów. Podstawowym problemem bezpieczeństwa przy szyfrowaniu symetrycznym pozostaje zatem skuteczne utajnienie używanych kluczy.

Przyjrzyjmy się dokładniej rysunkowi 2.2, na którym zilustrowano schemat szyfrowania symetrycznego. Generowany przez źródło komunikat, będący tekstem jawnym $X = [X_1, X_2, \ldots, X_M]$, składa się z M elementów będących znakami (literami) pewnego skończonego alfabetu. Tradycyjnie przyjmuje się w tej roli 26-literowy alfabet łaciński, choć większość współczesnych zastosowań opiera się na alfabecie bitowym (binarnym) `{0, 1}`. W celu zaszyfrowania tekstu jawnego należy wygenerować klucz $K = [K_1, K_2, \ldots, K_J]$. Jeśli klucz generowany jest w tym samym miejscu, co tekst jawny, pojawia się problem dostarczenia go odbiorcy za pośrednictwem bezpiecznego kanału komunikacyjnego. Alternatywą jest wygenerowanie klucza przez niezależny zaufany podmiot trzeci i bezpieczne dostarczenie go obu uczestnikom transmisji.

Traktując komunikat źródłowy X i klucz K jako informację wejściową dla algorytmu szyfrującego E, produkującego szyfrogram Y, możemy wyrazić powiązanie tych elementów w postaci wzoru

$$Y = \mathrm{E}(K, X)$$



Rysunek 2.2. Model szyfrowania symetrycznego

który można także rozumieć następująco: algorytm szyfrujący przekształca tekst jawny X na szyfrogram Y, a szczegóły tego przekształcenia parametryzowane są przez klucz K.

Uprawniony odbiorca komunikatu, dysponując kluczem K, potrafi odtworzyć komunikat X za pomocą algorytmu deszyfrującego D:

$$X = \mathrm{D}(K, Y)$$

Intruz (kryptoanalityk) dysponujący przechwyconym szyfrogramem Y, nie znający jednak X ani K, może podjąć próbę odtworzenia X i (lub) K. Zakładamy, że algorytmy E i D są powszechnie znane (również intruzowi). Jeżeli intruz zainteresowany jest odtworzeniem wyłącznie zaszyfrowanego tekstu jawnego X, jego wysiłki koncentrować się będą na konstruowaniu przybliżenia tego tekstu $\hat{X}$. Prawdopodobnie jednak intruz zainteresowany będzie także następnymi komunikatami, zmierzał więc będzie do konstruowania przybliżenia klucza $\hat{K}$.

Kryptografia

Każdy system kryptograficzny może być scharakteryzowany niezależnie pod względem każdego z trzech następujących kryteriów:

1. **Typu operacji przekształcających tekst jawny na szyfrogram**. Wszystkie algorytmy szyfrowania opierają się na dwojakiego typu operacjach: **podstawieniach**, w ramach których każdy element tekstu jawnego (bit, litera, grupa bitów, grupa liter) zastępowany jest przez inny element, oraz **przestawieniach** (transpozycjach), polegających za zmianie kolejności wspomnianych elementów. Wymaga się, aby operacje te były *odwracalne*, czyli

by szyfrowanie nie powodowało utraty informacji. Wiele systemów szyfrowania, zwanych *systemami produktowymi* (*product systems*) opiera się na skomplikowanych, wieloetapowych kombinacjach podstawień i przestawień.

2. **Liczby używanych kluczy**. W sytuacji, gdy nadawca i odbiorca współdzielą ten sam klucz, mówimy o szyfrowaniu symetrycznym, szyfrowaniu z pojedynczym kluczem, szyfrowaniu z kluczem tajnym lub szyfrowaniu konwencjonalnym (są to oczywiście synonimy). Gdy nadawca posługuje się kluczem innym niż odbiorca, mamy do czynienia z (ponownie synonimy) szyfrowaniem asymetrycznym, szyfrowaniem z dwoma kluczami lub szyfrowaniem z kluczami publicznymi.
3. **Sposobu przetwarzania tekstu jawnego**. *Szyfrowanie blokowe* polega na niezależnym przekształcaniu poszczególnych bloków tekstu jawnego w odpowiednie bloki szyfrogramu, podczas gdy *szyfrowanie strumieniowe* jest sukcesywnym tworzeniem pojedynczego strumienia szyfrogramu drogą przetwarzania kolejnych elementów tekstu jawnego, traktowanego także jako pojedynczy strumień[1].

Kryptoanaliza i atak siłowy

Zazwyczaj celem ataku na kryptosystem jest rozpoznanie wykorzystywanych kluczy, nie zaś rozpoznanie pojedynczego tekstu jawnego czy pojedynczego szyfrogramu. W przypadku szyfrowania konwencjonalnego do osiągnięcia tego celu wykorzystywane są głównie dwie następujące techniki:

- **Kryptoanaliza**. Technika ta bazuje na znajomości natury algorytmu szyfrującego i ewentualnie na pewnych ogólnych cechach tekstu jawnego bądź pary „tekst jawny – szyfrogram", a usiłowania stosującego ją intruza zmierzają w kierunku rozpoznania bądź to stosowanego klucza, bądź jedynie konkretnego tekstu jawnego.
- **Atak siłowy** (*brute-force*). Intruz dokonuje rozszyfrowywania szyfrogramu, wypróbowując wszystkie możliwe klucze, aż do uzyskania tekstu stwarzającego wrażenie (lub pozory) wiarygodności. W przeciętnym przypadku do osiągnięcia sukcesu konieczne jest wypróbowanie połowy wszystkich możliwych kluczy.

Gdy w wyniku którejkolwiek z tych technik uda się intruzowi wydedukować stosowany klucz, będzie on mógł odtwarzać z przechwytywanych szyfrogramów tekst jawny i całe szyfrowanie okaże się bezcelowe.

Zajmiemy się najpierw kryptoanalizą, po czym omówimy ataki siłowe.

W tabeli 2.1 widoczne jest zestawienie najczęściej stosowanych **ataków kryptoanalitycznych**, różniących się od siebie zestawem informacji, jaka dostępna jest intruzowi. Najtrudniejsza jest dla niego sytuacja, gdy dysponuje on *wyłącznie*

[1] W przeciwieństwie do szyfrowania blokowego przetwarzanie kolejnego elementu tekstu jawnego uzależnione jest od stanu określonego przez wynik przetwarzania poprzednich elementów — *przyp. tłum.*

Tabela 2.1. Rodzaje ataków kryptoanalitycznych

Rodzaj ataku	Informacja dostępna dla kryptoanalityka
Atak z samym szyfrogramem	• Algorytm szyfrujący • Przechwycony szyfrogram
Atak ze znanym tekstem jawnym	• Algorytm szyfrujący • Przechwycony szyfrogram • Jedna lub więcej par „tekst jawny – szyfrogram" utworzonych przy użyciu tego samego klucza
Atak z wybranym tekstem jawnym	• Algorytm szyfrujący • Przechwycony szyfrogram • Utworzony przez kryptoanalityka tekst jawny wraz z odpowiadającym mu szyfrogramem, utworzonym przy użyciu tajnego klucza
Atak z wybranym szyfrogramem	• Algorytm szyfrujący • Przechwycony szyfrogram • Utworzony przez kryptoanalityka szyfrogram wraz z odpowiadającym mu tekstem jawnym, odtworzonym przy użyciu tajnego klucza
Atak z wybranym tekstem jawnym i wybranym szyfrogramem	• Algorytm szyfrujący • Przechwycony szyfrogram • Utworzony przez kryptoanalityka tekst jawny wraz z odpowiadającym mu szyfrogramem, utworzonym przy użyciu tajnego klucza • Utworzony przez kryptoanalityka szyfrogram wraz z odpowiadającym mu tekstem jawnym, odtworzonym przy użyciu tajnego klucza

szyfrogramem i przypuszczalnie znajomością algorytmu deszyfrującego. Jedną z możliwości jest wówczas przypuszczenie przez niego ataku siłowego, jednakże w przypadku dużej liczby potencjalnie możliwych kluczy wypróbowanie ich wszystkich jest niewykonalne, a przynajmniej niepraktyczne. Alternatywą dla intruza jest więc wykorzystanie struktury szyfrogramu w drodze jego analizy statystycznej. Niezwykle pomocna może się wówczas okazać chociażby tylko ogólna wiedza na temat natury tekstu jawnego, który może być tekstem w języku angielskim czy francuskim, plikiem EXE, kodem programu w języku Java itp.

Najmniejsze szanse ma kryptoanalityk wówczas, gdy dysponuje tylko samym szyfrogramem — powodzenie ataku jest wówczas najmniej prawdopodobne. W wielu przypadkach kryptoanalityk dysponuje jednak bogatszą informacją, między innymi może przechwycić kilka próbek tekstu jawnego wraz z odpowiadającymi tym próbkom szyfrogramami. Użyteczna dla kryptoanalityka może być również informacja o ogólnych cechach tekstu jawnego: jeśli na przykład wiadomo, że przesyłany komunikat jest treścią pliku w formacie Postscript, to w ustalonych miejscach tego komunikatu można spodziewać się obecności pewnych ustalonych

wzorców i zasada ta pozostaje prawdziwa dla większości formatów danych. Dysponując wieloma egzemplarzami tekstu jawnego i wynikiem ich przekształcenia na szyfrogramy przy użyciu tego samego klucza, kryptoanalityk może podjąć próbę wydedukowania tego klucza — tego typu usiłowania określane są wspólnym mianem *ataku ze znanym tekstem jawnym*.

Pokrewna forma ataku opiera się na występowaniu w tekście jawnym określonych słów, dla których znane jest (być może w przybliżeniu) położenie w treści komunikatu. Nie ma oczywiście takiej regularności zwykły tekst prozatorski, ale na przykład w dokumencie opracowanym na zamówienie firmy X można spodziewać się informacji o prawach autorskich, obejmującej nazwę tej firmy, a strumień transmitujący kompletny plik danych księgowych przypuszczalnie zawiera na początku znany nagłówek.

Jeszcze więcej szczęścia ma analityk dysponujący możliwością „podłożenia" spreparowanego przez siebie tekstu jawnego na wejście systemu szyfrującego; obserwując postać szyfrogramów uzyskiwanych na podstawie dowolnie kształtowanego (a nie przechwytywanego) tekstu jawnego, kryptoanalityk może dedukować coraz trafniej postać klucza — na tej zasadzie opiera się kryptoanaliza różnicowa, o której piszemy w rozdziale 3. Ataki obejmujące możliwość dowolnego preparowania tekstu jawnego, który następnie poddawany jest szyfrowaniu, nazywane są ogólnie *atakami z wybranym tekstem jawnym*.

W analogiczny sposób może kryptoanalityk dokonywać deszyfracji dowolnie preparowanych przez siebie danych, traktowanych jako szyfrogramy (co nosi nazwę *ataków z wybranym szyfrogramem*), możliwe jest też połączenie obu sposobów. Tego typu ataki (odpowiadają im dwa ostatnie wiersze tabeli 2.1), choć stosowane rzadziej, wciąż są jednak możliwe.

Tylko kiepski algorytm szyfrujący daje kryptoanalitykowi duże szanse powodzenia w przypadku ataku ze znanym szyfrogramem (pierwszy wiersz tabeli 2.1). Generalnie od algorytmów szyfrujących wymaga się zdolności do skutecznego opierania się atakom ze znanym tekstem jawnym.

Warto w tym miejscu przedstawić dwie definicje związane z jakością algorytmu szyfrującego. Algorytm ten uważany jest za **bezwarunkowo bezpieczny**, jeśli generowane przezeń szyfrogramy nie zawierają informacji wystarczającej do jednoznacznego odtworzenia tekstu jawnego, niezależnie od tego jak obszerna jest baza tych szyfrogramów. Jak zobaczymy nieco później w tym rozdziale, poza schematem znanym jako „szyfrowanie z kluczem jednorazowym" nie istnieje algorytm szyfrujący spełniający ten warunek. Bezpieczeństwo teoretyczne musi zatem ustąpić miejsca podejściu praktycznemu — realne jest wymaganie spełnienia przez algorytm szyfrujący przynajmniej jednego z poniższych kryteriów:

- koszt złamania szyfru jest większy od wartości zaszyfrowanej informacji;
- czas potrzebny na złamanie szyfru przekracza okres użyteczności zaszyfrowanej informacji.

Gdy algorytm szyfrujący spełnia którykolwiek z powyższych kryteriów, nazywamy go **obliczeniowo bezpiecznym**. Niestety, oszacowanie czasochłonności i pracochłonności złamania danego szyfru jest zadaniem niezmiernie trudnym.

Wszystkie odmiany kryptoanalizy szyfrów symetrycznych wykorzystują prawdopodobieństwo, że pewna regularność tekstu otwartego znajduje swe odbicie w pewnych charakterystycznych cechach szyfrogramu. Zasada ta stanie się dla czytelników bardziej zrozumiała po omówieniu kilku przykładów szyfrowania symetrycznego. Dla odmiany kryptoanaliza ukierunkowana na szyfrowanie z kluczami publicznymi opiera się na próbie wydedukowania klucza prywatnego na podstawie zależności matematycznych wiążących go z kluczem publicznym.

Atak siłowy to mechaniczne wypróbowywanie kolejnych kluczy w nadziei, że któryś z nich zastosowany do przechwyconego szyfrogramu da w rezultacie tekst jawny przejawiający cechy autentyczności. W przeciętnym przypadku wymaga to sprawdzenia połowy wszystkich możliwych kluczy, co przy ich dużej liczbie może dyskwalifikować całe przedsięwzięcie już na starcie (no, może niekoniecznie, wytrwałym szczęście sprzyja — teoretycznie możliwe jest, że już pierwszy sprawdzony klucz okaże się tym właściwym; zaufajmy jednak statystyce). W tabeli 2.2 widoczny jest rozmiar czasochłonności łamania szyfrów z kluczami różnych rozmiarów. Klucz 56-bitowy wykorzystywany jest w algorytmie DES (*Data Encryption Standard*), z klucza 168-bitowego korzysta natomiast algorytm „potrójnego DES" (*Triple DES*). Algorytm AES (*Advanced Encryption Standard*) opiera się na kluczu 128-bitowym. Uwzględniono także klucz złożony z 26 liter alfabetu w określonym ich uporządkowaniu. Dla każdego klucza podano przybliżony czas realizacji ataku siłowego, prowadzonego przy wykorzystywaniu dwóch różnych rzędów mocy obliczeniowych. Przeprowadzenie pojedynczego dekryptażu w czasie 1 mikrosekundy (10^{-6} s) odpowiada w przybliżeniu możliwościom dzisiejszych komputerów; przy zastosowaniu masowego zrównoleglenia w systemach wieloprocesorowych można jednak uzyskiwać rezultaty lepsze o kilka rzędów wielkości — w ostatniej kolumnie tabeli podana jest czasochłonność łamania szyfru dla systemu zdolnego sprawdzić milion kluczy w ciągu mikrosekundy. I jak łatwo zauważyć, przy tej mocy obliczeniowej algorytm DES nie może być uważany za obliczeniowo bezpieczny.

Tabela 2.2. Średni czas potrzebny do wyczerpującego przeszukania przestrzeni możliwych kluczy

Rozmiar klucza	Liczba możliwych kluczy	Czas potrzebny w systemie sprawdzającym jeden klucz w ciągu μs	Czas potrzebny w systemie sprawdzającym milion kluczy w ciągu μs
32 bity	$2^{32} \approx 4{,}3 \times 10^9$	2^{31} μs ≈ 35,8 minuty	2,15 milisekundy
56 bitów	$2^{56} \approx 7{,}2 \times 10^{16}$	2^{55} μs ≈ 1142 lata	10,01 godziny
128 bitów	$2^{128} \approx 3{,}4 \times 10^{38}$	2^{127} μs ≈ $5{,}4 \times 10^{24}$ lat	$5{,}4 \times 10^{18}$ lat
168 bitów	$2^{168} \approx 3{,}7 \times 10^{50}$	2^{167} μs ≈ $5{,}9 \times 10^{36}$ lat	$5{,}4 \times 10^{30}$ lat
26 znaków (w dowolnym uporządkowaniu)	$26! \approx 4 \times 10^{26}$	2×10^{26} μs ≈ $6{,}4 \times 10^{12}$ lat	$6{,}4 \times 10^{6}$ lat

2.2. TECHNIKI PODSTAWIENIOWE

W tej i następnej sekcji przedstawimy przykłady kilku technik, które z racji historycznego znaczenia można uznać za klasyczne techniki szyfrowania. Ich dokładne przeanalizowanie pozwoli lepiej zrozumieć obecne podejście do szyfrowania symetrycznego, a także rozpoznać potencjalne typy możliwych ataków, którym projektanci schematów szyfrowania muszą się a priori przeciwstawić.

Dwoma fundamentami każdej techniki szyfrowania są podstawienia i przestawienia. Prześledzimy je dokładnie w dwóch kolejnych sekcjach, po czym zaprezentujmy metodę szyfrowania łączącą przestawienia z podstawieniami.

Podstawianiem (*substitution*) nazywamy technikę, w ramach której kolejne litery tekstu jawnego zastępowane są określonymi literami, liczbami i symbolami. Jeśli tekst jawny rozpatrywany jest jako ciąg bitów, wyodrębniane z niego kolejne wzorce bitowe zastępowane są wzorcami bitowymi tworzącymi szyfrogram.

W dalszym ciągu książki dla polepszenia czytelności tekst jawny zapisywać będziemy przy użyciu `małych liter`, zaś szyfrogram — przy użyciu `DUŻYCH LITER`. Klucze zapisywać będziemy *`małymi pochylonymi literami`*.

Szyfr Cezara

Najstarsze znane, i zarazem najprostsze, zastosowanie techniki podstawieniowej datuje się z czasów Juliusza Cezara. Szyfrowanie Cezara polega na zastępowaniu każdej litery tekstu jawnego literą znajdującą się trzy pozycje dalej w alfabecie, co zapisać można bardzo prosto w następującej postaci:

```
tekst jawny: a b c d e f g h i j k l m n o p q r s t u v w x y z
szyfrogram:  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Zauważmy, że określenie „trzy pozycje dalej" należy rozumieć w sensie cyklicznym — trzy ostatnie litery alfabetu zastępowane są trzema pierwszymi.

Tak więc przykładowy komunikat zostanie zaszyfrowany następująco:

```
tekst jawny: meet me after the toga party
szyfrogram:  PHHW PH DIWHU WKH WRJD SDUWB
```

Przyporządkujmy każdej literze odpowiednik liczbowy:

a	b	c	d	e	f	g	h	i	j	k	l	m
0	1	2	3	4	5	6	7	8	9	10	11	12

n	o	p	q	r	s	t	u	v	w	x	y	z
13	14	15	16	17	18	19	20	21	22	23	24	25

Wówczas algorytm realizujący szyfrowanie Cezara będzie można formalnie zapisać następująco:

każdą literę p tekstu jawnego zastępujemy literą szyfrogramu C wyznaczoną według formuły[2]

$$C = \mathrm{E}(3, p) = (p + 3) \bmod 26$$

Regułę tę można w oczywisty sposób uogólnić na dowolną wartość przesunięcia, traktowanego jako klucz:

$$C = \mathrm{E}(k, p) = (p + k) \bmod 26 \qquad \textbf{(2.1)}$$

gdzie k może przyjmować wartości od 1 do 25. Z formuły (2.1) wynika formuła algorytmu deszyfrującego:

$$p = \mathrm{D}(k, C) = (C - k) \bmod 26 \qquad \textbf{(2.2)}$$

Ponieważ k może przyjmować tylko 25 różnych wartości, więc kryptoanalityk z powodzeniem zastosować może atak siłowy. Potencjalny efekt jego eksperymentów uwidoczniony jest na rysunku 2.3 — jedynie w trzecim wierszu znajduje się sensowny tekst.

```
        PHHW PH DIWHU WKH WRJD SDUWB
KLUCZ
    1   oggv og chvgt vjg vqic rctva
    2   nffu nf bgufs uif uphb qbsuz
    3   meet me after the toga party
    4   ldds ld zesdq sgd snfz ozqsx
    5   kccr kc ydrcp rfc rmey nyprw
    6   jbbq jb xcqbo qeb qldx mxoqv
    7   iaap ia wbpan pda pkcw lwnpu
    8   hzzo hz vaozm ocz ojbv kvmot
    9   gyyn gy uznyl nby niau julns
   10   fxxm fx tymxk max mhzt itkmr
   11   ewwl ew sxlwj lzw lgys hsjlq
   12   dvvk dv rwkvi kyv kfxr grikp
   13   cuuj cu qvjuh jxu jewq fqhjo
   14   btti bt puitg iwt idvp epgin
   15   assh as othsf hvs hcuo dofhm
   16   zrrg zr nsgre gur gbtn cnegl
   17   yqqf yq mrfqd ftq fasm bmdfk
   18   xppe xp lqepc esp ezrl alcej
   19   wood wo kpdob dro dyqk zkbdi
   20   vnnc vn jocna cqn cxpj yjach
   21   ummb um inbmz bpm bwoi xizbg
   22   tlla tl hmaly aol avnh whyaf
   23   skkz sk glzkx znk zumg vgxze
   24   rjjy rj fkyjw ymj ytlf ufwyd
   25   qiix qi ejxiv xli xske tevxc
```

Rysunek 2.3. Złamanie szyfru Cezara metodą ataku siłowego

[2] Zapis `a mod b` oznacza resztę z dzielenia `a` przez `b`, na przykład `11 mod 7 = 4`.

Zauważmy, że powodzenie ataku siłowego wynika w tym przypadku z trzech następujących faktów:

1. Znane są algorytmy szyfrujący i deszyfrujący.
2. Liczba możliwych kluczy jest niewielka.
3. Język, w jakim zapisano tekst jawny, jest znany i łatwo rozpoznawalny.

W większości sytuacji wziętych z codziennej rzeczywistości spełniony jest warunek 1., o czym pisaliśmy już wcześniej w tym rozdziale. Tym, co generalnie skazuje na niepowodzenie większość ataków siłowych, jest niespełnienie warunku 2.: w przypadku szyfru „potrójnego DES" (opisywanego w rozdziale 6.) liczba możliwych kluczy wynosi $2^{168} \approx 3{,}7 \times 10^{50}$.

Nie bez znaczenia jest także trzecia z wymienionych przesłanek. Jeżeli język tekstu jawnego nie jest znany, trafienie na właściwy klucz może zostać po prostu niezauważone. Ponadto sam tekst jawny może zostać przed zaszyfrowaniem poddany kompresji lub innym przekształceniom redukującym, co dodatkowo utrudnia jego zrozumienie. Na rysunku 2.4 widoczny jest fragment archiwum .ZIP w oknie edytora tekstowego. Gdyby zbiór ten został zaszyfrowany zwykłą techniką podstawieniową, fakt jego rozszyfrowania za pomocą ataku siłowego mógłby pozostać niezauważony[3].

Rysunek 2.4. Fragment (znakowej) zawartości skompresowanego pliku

Szyfry monoalfabetyczne

Z 25-elementową przestrzenią kluczy szyfr Cezara zdecydowanie nie zasługuje na miano bezpiecznego. Drastyczne zwiększenie liczebności tej przestrzeni można uzyskać, komplikując reguły podstawiania. Dla danego skończonego zbioru *S*

[3] Ale uwaga: pliki takie jak archiwa cechują się pewnym stopniem redundancji. Aby archiwum zostało poprawnie obsłużone przez archiwizator, musi m.in. posiadać poprawny nagłówek, poprawne sumy kontrolne itp. Nieoczekiwanie fakt skompresowania tekstu otwartego archiwizatorem ZIP może ułatwić zadanie kryptoanalitykowi, daje mu bowiem bardzo wygodne narzędzie weryfikowania trafności wyboru klucza: po deszyfracji należy po prostu uruchomić program PKZIP (oczywiście, o ile wie, że zaszyfrowanym plikiem jest archiwum ZIP) — udane uruchomienie archiwizatora (kod powrotu 0) oznacza prawdopodobne trafienie we właściwy klucz — *przyp. tłum.*

definiujemy jego **uporządkowanie**[4] jako dowolny ciąg, w którym każdy element tego zbioru występuje dokładnie raz. Przykładowo: dla zbioru $S = \{a, b, c\}$ istnieje sześć różnych uporządkowań: `abc`, `acb`, `bac`, `bca`, `cab`, `cba`. Ogólnie dla n-elementowego zbioru istnieje dokładnie $n!$ różnych uporządkowań, ponieważ pierwszy element ciągu wybrać można na n sposobów, drugi — na $(n - 1)$ sposobów, trzeci — na $(n - 2)$ sposoby itd., wobec czego liczba możliwych wariantów wyboru wynosi

$$n \times (n - 1) \times (n - 2) \times \ldots \times 2 \times 1 = n!$$

Przywołajmy ponownie szyfr Cezara:

```
tekst jawny: a b c d e f g h i j k l m n o p q r s t u v w x y z
szyfrogram:  D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

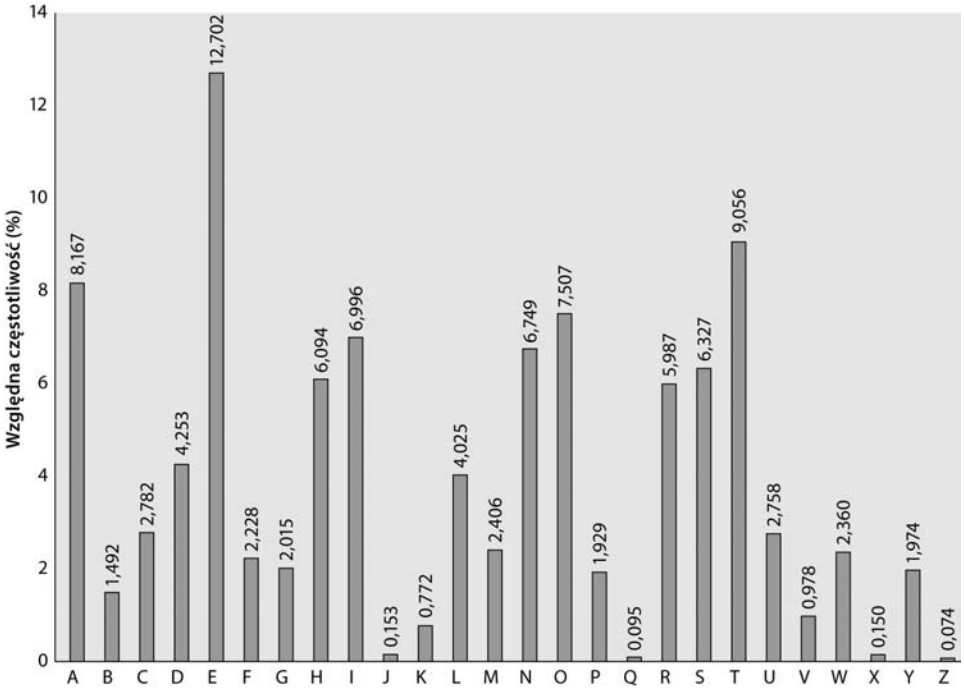
jeżeli dopuścimy, by wiersz „szyfrogram" zawierał dowolne uporządkowanie zbioru 26 liter, liczba możliwych kluczy zwiększa się do wartości $26! \approx 4 \times 10^{26}$. Daje to przestrzeń o liczebności 10 rzędów większej niż liczba możliwych kluczy szyfru DES, co powinno zdecydowanie podziałać zniechęcająco na amatora ataku siłowego. Ponieważ (dla każdego konkretnego uporządkowania) reguły zastępowania są ustalone dla całego komunikatu — nie zmienia się „alfabet kodowy" w wierszu szyfrogram — ten rodzaj podstawiania nazywamy **szyfrowaniem monoalfabetycznym**.

Co prawda 27-cyfrowa liczba możliwych kluczy do sprawdzenia dyskwalifikuje na starcie próby ataku siłowego, lecz kryptoanalityk dysponuje ponadto innymi, bardziej obiecującymi metodami. Jeśli mianowicie zna on naturę tekstu jawnego (na przykład wie, że jest to zwykły tekst w języku angielskim), może wykorzystywać pewne cechy charakterystyczne dla tegoż języka. Weźmy przykładowy szyfrogram, zaczerpnięty z książki [SINK66]:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
```

Dla szyfrogramu tego można wykonać analizę częstości występowania poszczególnych liter i skonfrontować jej wyniki ze statystyką występowania poszczególnych liter w zwykłym tekście anglojęzycznym (histogram widoczny na rysunku 2.5 zaczerpnięty został z pracy [LEWA00]). Dla dostatecznie długiego komunikatu konfrontacja ta może okazać się wystarczająca; w naszym przykładzie komunikat jest stosunkowo krótki, więc można się wpierw spodziewać jedynie przybliżonego wyniku. Rozkład (w procentach) częstości występowania poszczególnych liter w szyfrogramie wygląda tak:

[4] W amerykańskim wydaniu tej książki opisane przekształcenie zbioru elementów na ich ciąg nazywane jest **permutacją**. Może to być mylące, w algebrze bowiem „permutacja" definiowana jest jako wzajemnie jednoznaczne przekształcenie danego zbioru na siebie. By uniknąć wynikającej stąd dwuznaczności, stosowany będzie w zamian termin „uporządkowanie" — *przyp. tłum.*



Rysunek 2.5. Względna częstotliwość występowania poszczególnych liter w tekstach w języku angielskim[5]

P 13,33	H 5,83	F 3,33	B 1,67	C 0,00
Z 11,67	D 5,00	W 3,33	G 1,67	K 0,00
S 8,33	E 5,00	Q 2,50	Y 1,67	L 0,00
U 8,33	V 4,17	T 2,50	I 0,83	N 0,00
O 7,50	X 4.17	A 1,67	J 0,83	R 0,00
M 6,67				

Porównanie obu statystyk sugeruje, że litery P i Z, jako występujące w szyfrogramie najczęściej, odpowiadają literom e i t tekstu jawnego, chociaż nie wiadomo jeszcze, która jest która. Litery S, U, O, M i H o względnie dużej częstotliwości odpowiadają literom ze zbioru [a, h, i, n, o, r, s]. Litery występujące w szyfrogramie najrzadziej — A, B, G, Y, I, J — mają najprawdopodobniej swe odpowiedniki w zbiorze [b, j, k, q, v, x, z].

Posiadając tę wiedzę, moglibyśmy metodą prób i błędów dojść do jakiegoś sensownego „szkieletu" poszukiwanego komunikatu, możliwe jest jednak inne, bardziej systematyczne podejście, eksploatujące inne regularności typowe dla języka angielskiego, na przykład duże prawdopodobieństwo obecności określonych słów w każdym niemal tekście czy też powtarzające się sekwencje liter mające swe odzwierciedlenie w podobnych powtórzeniach w ramach szyfrogramu.

[5] Częstotliwość występowania poszczególnych liter w tekstach w języku polskim można znaleźć na stronie: *http://pl.wikipedia.org/wiki/Alfabet_polski*

Powróćmy jednak do statystyki. Wykres podobny do widocznego na rysunku 2.5 można sporządzić również w odniesieniu do **dwuznaków** (zwanych także *digramami*) — w języku angielskim najczęściej występującym dwuznakiem jest `th`. W naszym szyfrogramie najczęściej powtarzającym się jest dwuznak `ZW`, który występuje trzykrotnie. Wnioskujemy stąd, że `Z` odpowiada literze `t`, zaś `W` — literze `h`. Wcześniejsze spostrzeżenie dotyczące liter `P` i `Z` pozwala nam na stwierdzenie, że `P` odpowiada literze `e`. Zatem sekwencja `ZWP` w szyfrogramie odpowiada sekwencji `the` w tekście jawnym. To najczęściej występujący w języku angielskim trójznak (*trigram*), co pozwala nam sądzić, że jesteśmy na dobrej drodze.

Zwróćmy następnie uwagę na sekwencję `ZWSZ` w pierwszym wierszu. Nie ma pewności, że odpowiada ona pełnemu słowu, ale jeśli tak jest, słowo to musi mieć postać `th_t`, prawdopodobnie więc odpowiednikiem `S` jest `a`.

Reasumując, dotychczas otrzymaliśmy następujący wynik:

```
UZQSOVUOHXMOPVGPOZPEVSGZWSZOPFPESXUDBMETSXAIZ
 t a        e  e te  a that e e a       a   t
VUEPHZHMDZSHZOWSFPAPPDTSVPQUZWYMXUZUHSX
   e t   ta t ha e ee  a e  th    t  a
EPYEPOPDZSZUFPOMBZWPFUPZHMDJUDTMOHMQ
 e  e e tat  e   the   t
```

Mimo iż udało nam się odgadnąć tylko cztery litery tekstu jawnego, uzyskaliśmy już dość znaczną część całego tekstu. Kontynuując dedukcję metodą prób i błędów, a na końcu wstawiając spacje rozdzielające słowa, otrzymamy ostatecznie test otwarty:

```
it was disclosed yesterday that several informal but
direct contacts have been made with political
representatives of the viet cong in moscow
```

Szyfry monoalfabetyczne poddają się kryptoanalizie o tyle łatwo, że w szyfrogramie odzwierciedlona zostaje statystyka wystąpień poszczególnych znaków i ich sekwencji w tekście jawnym. Tę odpowiedniość można jednak wyraźnie osłabić, stosując dla określonej litery tekstu jawnego *kilka* różnych odpowiedników zwanych **homofonami**, na przykład zastępując literę **e** jedną z liczb `16`, `74`, `35` i `21` — przyporządkowanie kolejnych homofonów danej literze tekstu jawnego może zmieniać się cyklicznie lub w sposób losowy. Jeżeli liczba homofonów odpowiadających danej literze będzie tym większa, im częściej litera ta występuje w tekście jawnym, to statystyka wystąpień poszczególnych znaków w tekście jawnym zatraci się prawie kompletnie w szyfrogramie. Pomysłodawca tej zasady, wielki matematyk Carl redrich Gauss wierzył, że wykorzystywanie homofonów uczyni szyfrowanie praktycznie niemożliwym do złamania. Jednak nawet użycie homofonów nie zmienia faktu, że jeden znak tekstu otwartego odwzorowywany jest na jeden znak szyfrogramu, i statystyka wystąpień sekwencji znaków (między innymi dwu- i trójznaków) zachowywana jest w szyfrogramie.

W celu osłabienia stopnia, w jakim rozmaite statystyki tekstu jawnego odzwierciedlane są w szyfrogramie, zaproponowano dwa podejścia: rozpatrywanie tekstu

jawnego w podziale nie na pojedyncze litery, a na jednostki wieloliterowe oraz wykorzystywanie wielu alfabetów szyfrogramu zamiast pojedynczego alfabetu. Omówimy krótko każde z nich.

Szyfr Playfaira

Najbardziej znanym przykładem rozpatrywania tekstu jawnego w podziale na sekwencje wieloznakowe jest szyfr Playfaira[6], w którym tekst jawny dzielony jest na dwuznaki.

Podstawą szyfru jest macierz znaków o rozmiarze 5×5. W alfabecie utożsamiamy ze sobą litery I oraz J, dzięki czemu liczba liter redukuje się do 25. Ich określone rozmieszczenie we wspomnianej macierzy jest kluczem szyfru. Rozpoczynamy od wyboru słowa kluczowego: nie może ono zawierać powtórzeń liter, a jeżeli powtórzenia jednak występują, należy je wyeliminować — i tak na przykład słowo *jaundicing* zredukowane zostanie w wyniku tego zabiegu do słowa *jaundcg* (pamiętajmy o utożsamieniu *i* oraz *j*). Słowo kluczowe wpisujemy do macierzy, poruszając się po kolejnych wierszach od lewej do prawej, z góry na dół. Następnie wpisujemy w kolejności alfabetycznej pozostałe litery nieobecne w słowie kluczowym. Dla słowa kluczowego *monarchy* macierz Playfaira przyjmie zatem postać następującą[7]:

M	O	N	A	R
C	H	Y	B	D
E	F	G	I/J	K
L	P	Q	S	T
U	V	W	X	Z

Szyfrowanie tekstu jawnego odbywa się kolejnymi dwuznakami. Każdy dwuznak musi zawierać różne znaki; jeśli postać tekstu jawnego prowadzi do naruszenia tej zasady, należy sztucznie rozdzielić bliźniacze znaki jakimś „neutralnym" znakiem, tak by nie powodowało to zmiany znaczenia tekstu. Dla tekstu jawnego `balloon`, używając w tym celu litery `x`, otrzymujemy podział na dwuznaki `ba lx lo on`. Każdy dwuznak tekstu jawnego odwzorowywany jest na dwuznak szyfrogramu, przy czym w zależności od wzajemnego położenia składników dwuznaku w macierzy Playfaira wyróżniamy trzy przypadki:

1. Gdy oba składniki leżą w tym samym wierszu macierzy, zastępujemy każdy z nich prawostronnym następnikiem; prawostronnym następnikiem dla ostatniego znaku w wierszu jest pierwszy znak tego wiersza. Formalnie

[6] Opisywany szyfr wynaleziony został w 1854 roku przez Charlesa Wheatstone'a, swoją nazwę zawdzięcza jednak baronowi Lyonowi Playfairowi, szkockiemu naukowcowi i parlamentarzyście, który w latach 1873 – 1874 pełnił ministerialną funkcję generalnego poczmistrza.

[7] Przykład pochodzi z książki Dorothy Sayers *Have His Carcase*.

zatem rzecz biorąc, dwuznak $P_{ij}P_{im}$ (P oznacza macierz Playfaira) odwzorowany zostaje na dwuznak $P_{i,next(j)}P_{i,next(m)}$, gdzie *next()* jest funkcją następnika:

$$next(i) = \begin{cases} i+1 \text{ dla } i < 5 \\ 1 \quad \text{dla } i = 5 \end{cases}$$

Zatem na przykład dwuznak `ar` odwzorowywany jest na dwuznak `RM`.

2. Analogicznie gdy oba składniki leżą w tej samej kolumnie macierzy, zastępujemy każdy z nich dolnym następnikiem; dla ostatniego wiersza wierszem następnym jest pierwszy wiersz. Formalnie dwuznak $P_{ij}P_{kj}$ odwzorowany zostaje na dwuznak $P_{next(i),j}P_{next(k),j}$, na przykład dwuznak `mu` odwzorowywany jest na dwuznak `CM`.
3. W pozostałym przypadku, to znaczy w sytuacji, gdy składniki dwuznaku leżą w różnych wierszach i różnych kolumnach, stosujemy tzw. uzupełnienie prostokątne: obrazem składnika jest element leżący w tym samym wierszu i kolumnie partnera. Formalnie dwuznak $P_{ij}P_{km}$ odwzorowany zostaje na dwuznak $P_{im}P_{kj}$, na przykład dwuznak `bp` odwzorowywany jest na dwuznak `IM` (albo `JM`, do wyboru)[8].

Szyfr Playfaira ma ogromną przewagę nad prostymi szyframi monoalfabetycznymi między innymi z tego względu, że zamiast 26 znaków mamy 26×26 = 676 dwuznaków, których indywidualna identyfikacja staje się z tego powodu znacznie utrudniona. Ponadto względne częstotliwości występowania poszczególnych liter wykazują znaczne zróżnicowanie w porównaniu z rozkładem częstotliwości dwuznaków, co czyni kryptoanalizę jeszcze trudniejszą. Z tego powodu szyfr Playfaira uważany był przez długi czas za wyjątkowo bezpieczny; wykorzystywany był w czasie I wojny światowej przez armię brytyjską oraz przez armię USA i aliantów podczas II wojny.

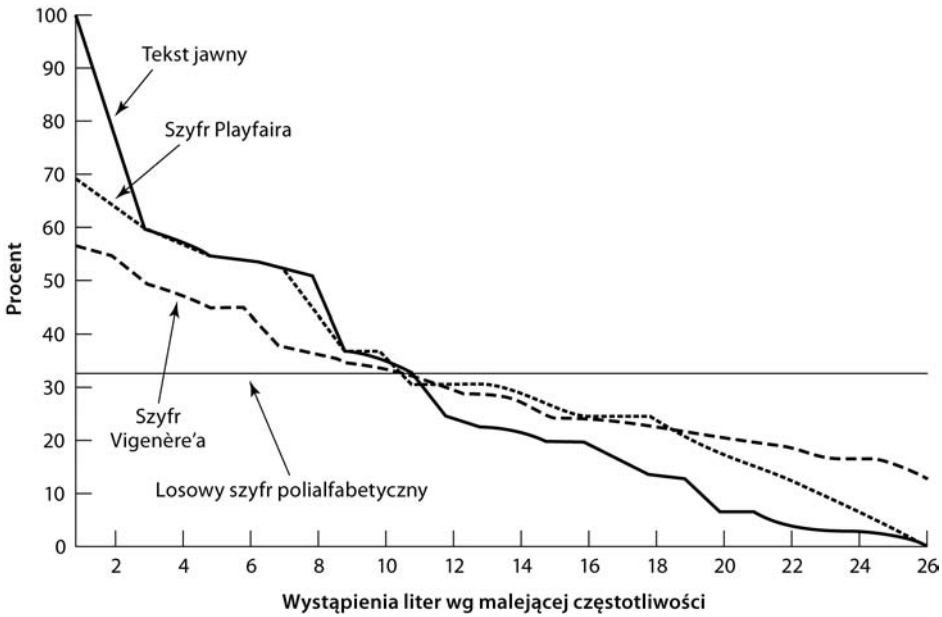
Pomimo jednak tak wielkiego zaufania szyfr Playfaira okazuje się niezbyt trudny do złamania, ponieważ produkowane przy jego użyciu szyfrogramy wciąż zachowują wiele cech struktury używanego języka; w praktyce kilkaset liter szyfrogramu okazuje się wystarczające do odtworzenia tekstu jawnego.

Jeden ze sposobów oceny efektywności szyfru Playfaira i innych szyfrów opiera się na specyficznej analizie częstotliwości wystąpień poszczególnych liter, której rezultaty widoczne są na rysunku 2.6, zaczerpniętym z pracy [SIMM93][9].

[8] Zobaczmy przy okazji, skąd — w kontekście powyższych reguł — bierze się konieczność unikania bliźniaczych dwuznaków w tekście jawnym. Gdybyśmy takowe dopuścili (załóżmy dla ustalenia uwagi, że wejściowym dwuznakiem jest `FF`), można by dla nich zastosować regułę 1. (dla `FF` otrzymalibyśmy wtedy wynik `GG`) albo regułę 2. (co dla `FF` dałoby wynik `PP`). I pojawiłaby się niejednoznaczność, bo napotykając w szyfrogramie bliźniaczy dwuznak, nie potrafilibyśmy określić, na mocy której reguły został on utworzony: `PP` mogłoby być obrazem zarówno `LL` (reguła 1.), jak i `FF` (reguła 2.). Oczywiście zdefiniowanie dodatkowej reguły mogłoby rozwiązać ten problem, jednak tak czy inaczej, obrazem bliźniaczego dwuznaku w tekście jawnym musiałby być bliźniaczy dwuznak szyfrogramie (co okazuje się oczywiste, gdy próbuje się zaprzeczyć temu stwierdzeniu), a to mogłoby być dla potencjalnego kryptoanalityka znacznym ułatwieniem — *przyp. tłum.*

[9] Dziękuję Gustavusowi Simmonsowi za udostępnienie wykresu i wyjaśnienie metody jego sporządzenia.

Linia zatytułowana „tekst jawny" odzwierciedla rozkład częstotliwości poszczególnych liter w tekście liczącym ich ponad 70 000, składających się na artykuł o kryptografii w *Encyclopaedia Britannica*. Liczby na osi poziomej reprezentują poszczególne litery uszeregowane w kolejności malejących częstości występowania — dla każdego z rozważanych szyfrów odpowiedniość między liczbami a literami jest oczywiście inna. Konkretna postać tej odpowiedniości nie jest zresztą istotna, bo znacznie ważniejsze jest tu co innego, a mianowicie zróżnicowanie wspomnianej częstotliwości dla poszczególnych szyfrogramów. Szyfrogramy te utworzono ze wspomnianego tekstu jawnego przy użyciu różnych algorytmów szyfrujących, a wykresy ilustrujące dystrybucję poszczególnych liter sporządzono, zliczając ich wystąpienia i dzieląc otrzymane liczniki przez liczbę wystąpień litery e (jako najczęstszej) w tekście jawnym. Dla tekstu jawnego otrzymujemy więc wskaźnik 100% dla litery e, ok. 76% dla litery a itd.



Rysunek 2.6. Względna częstotliwość występowania liter w tekście jawnym i różnych szyfrogramach

Dzięki opisanej normalizacji wykres widoczny na rysunku 2.6 może stanowić znakomity przyczynek do dyskusji o tym, w jakim stopniu szyfrowanie jest w stanie zamaskować dystrybucję występowania poszczególnych liter w tekście jawnym. Jeśli dla danego szyfru maskowanie to ma rozmiar totalny, wykres dystrybucji znaków dla tego szyfru powinien być płaską linią — kryptoanaliza jest wtedy praktycznie niemożliwa. Jak widać, dla szyfru Playfaira wspomniana linia jest bardziej spłaszczona niż w przypadku tekstu jawnego, mimo to szyfr ten odzwierciedla w znaczącym stopniu oryginalną dystrybucję poszczególnych znaków.

Szyfr Hilla

Inny interesujący szyfr wieloliterowy wynaleziony został w 1929 roku przez matematyka Lestera Hilla. Szyfr ten bazuje na arytmetyce macierzowej modulo 26, dlatego na wstępie przypomnimy kilka niezbędnych faktów z zakresu algebry liniowej; czytelnika zainteresowanego szczegółami mnożenia i odwracania macierzy odsyłamy do dodatku E[10].

Podstawowe koncepcje arytmetyki macierzowej

Macierzą odwrotną do macierzy kwadratowej **M**, oznaczaną $\mathbf{M}^{-1}$, jest macierz spełniająca warunek $\mathbf{M}(\mathbf{M}^{-1}) = \mathbf{M}^{-1}\mathbf{M} = \mathbf{I}$, gdzie **I** jest macierzą jednostkową, czyli macierzą posiadającą jedynki na głównej przekątnej i zera poza nią. Nie dla każdej macierzy istnieje macierz odwrotna, ale jeśli istnieje, spełnia wspomniany warunek. Przykładowo

$$\mathrm{A} = \begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix} \qquad \mathrm{A}^{-1} \bmod 26 = \begin{bmatrix} 9 & 2 \\ 1 & 15 \end{bmatrix}$$

$$\mathrm{AA}^{-1} = \begin{bmatrix} (5\times 9)+(8\times 1) & (5\times 2)+(8\times 15) \\ (17\times 9)+(3\times 1) & (17\times 2)+(3\times 15) \end{bmatrix}$$

$$= \begin{bmatrix} 53 & 130 \\ 156 & 79 \end{bmatrix} \bmod 26 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

Dla wyjaśnienia, jak oblicza się macierz odwrotną, zdefiniujemy pojęcie **wyznacznika** (ang. *determinant*). Dla macierzy kwadratowej tworzymy wszelkie możliwe iloczyny jej elementów, takie że każdy element pochodzi z innego wiersza i z innej kolumny. Zależnie od wzajemnego układu wierszy i kolumn elementów tworzących dany iloczyn zmieniamy jego znak lub pozostawiamy znak bez zmiany. Tak otrzymane $n!$ iloczynów (n jest wymiarem macierzy) dodajemy do siebie — otrzymana suma jest rzeczonym wyznacznikiem; wyznacznik macierzy **M** oznaczamy `det` **M**. Przykładowo dla macierzy o wymiarze 2×2

$$\begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix}$$

wyznacznik równy jest $k_{11}k_{22} - k_{12}k_{21}$, zaś dla macierzy rozmiaru 3×3 jest on równy $k_{11}k_{22}k_{33} + k_{21}k_{32}k_{13} + k_{31}k_{12}k_{23} - k_{31}k_{22}k_{13} - k_{21}k_{12}k_{33} - k_{11}k_{32}k_{23}$. Macierz, której wyznacznik jest zerowy, nazywa się **macierzą osobliwą**. Niezerowość wyznacznika, czyli nieosobliwość macierzy jest warunkiem koniecznym i wystarczającym istnienia macierzy do niej odwrotnej. Elementy macierzy odwrotnej obliczamy z równości

$$[\mathbf{A}^{-1}]_{ij} = (\det \mathbf{A})^{-1}(-1)^{i+j}(\mathrm{D}_{ij})$$

[10] Ten szyfr jest być może nieco trudniejszy do zrozumienia niż inne szyfry opisywane w tym rozdziale, ilustruje jednak pewną istotną kwestię związaną z kryptoanalizą. Przy pierwszym czytaniu można tę podsekcję pominąć.

gdzie D_{ij} jest podwyznacznikiem, czyli wyznacznikiem macierzy powstającej z macierzy **A** przez usunięcie jej *i*-tego wiersza i *j*-tej kolumny. $(\det \mathbf{A})^{-1}$ oznacza multiplikatywną odwrotność wyznacznika det **A** modulo 26

$$(\det \mathbf{A})^{-1} \times (\det \mathbf{A}) \bmod 26 = 1$$

Powracając do naszego przykładu

$$\det\begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix} = (5\times 3) - (8\times 17) = -121 \bmod 26 = 9$$

Liczba 3 jest multiplikatywną odwrotnością liczby 9, ponieważ (3×9) mod 26 = 27 mod 26 = 1 (patrz rozdział 4. lub dodatek E). Zatem macierzą odwrotną modulo 26 do macierzy

$$\mathrm{A} = \begin{bmatrix} 5 & 8 \\ 17 & 3 \end{bmatrix}$$

jest macierz[11]

$$\mathrm{A}^{-1} \bmod 26 = 3\begin{bmatrix} 3 & -8 \\ -17 & 5 \end{bmatrix} = 3\begin{bmatrix} 3 & 18 \\ 9 & 5 \end{bmatrix} = \begin{bmatrix} 9 & 54 \\ 27 & 15 \end{bmatrix} = \begin{bmatrix} 9 & 2 \\ 1 & 15 \end{bmatrix}$$

ALGORYTM HILLA

Algorytm szyfrujący Hilla traktuje tekst jawny jako ciąg *m*-literowych sekwencji znaków — każda z tych sekwencji przekształcana jest na *m*-literową sekwencję szyfrogramu. Przekształcenie to realizowane jest przez *m* równań liniowych, w których zarówno litery tekstu jawnego, jak i litery szyfrogramu traktowane są jako wartości numeryczne (a = 0, b = 1, …, z = 25); każde równanie określa jeden znak szyfrogramu. Przykładowo dla *m* = 3 mamy

$$\begin{aligned} c_1 &= (k_{11}p_1 + k_{12}p_2 + k_{13}p_3) \bmod 26 \\ c_2 &= (k_{21}p_1 + k_{22}p_2 + k_{23}p_3) \bmod 26 \\ c_3 &= (k_{31}p_1 + k_{32}p_2 + k_{33}p_3) \bmod 26 \end{aligned}$$

co w zapisie macierzowym prezentuje się następująco[12]:

$$\begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}^{\mathrm{T}} \bmod 26$$

[11] Znaki równości oznaczają tu przystawanie modulo 26 — *przyp. tłum.*

[12] W niektórych książkach poświęconych kryptografii zarówno tekst jawny, jak i szyfrogram reprezentowane są w postaci wektorów kolumnowych, jako takich mnożonych lewostronnie przez macierz (w przeciwieństwie do wektorów wierszowych mnożonych prawostronnie). Przyjęliśmy konwencję wektorów wierszowych, taka bowiem stosowana jest w systemie Sage. W rezultacie jednak zwyczajowe mnożenie

$\begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix} \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$ przyjmuje teraz postać $\begin{bmatrix} p_1 & p_2 & p_3 \end{bmatrix} \begin{bmatrix} k_{11} & k_{12} & k_{13} \\ k_{21} & k_{22} & k_{23} \\ k_{31} & k_{32} & k_{33} \end{bmatrix}^{\mathrm{T}}$.

lub krócej

$$\mathbf{C} = \mathbf{PK} \bmod 26$$

gdzie **C** i **P** są wektorami wierszowymi o rozmiarze 3, reprezentującymi (odpowiednio) szyfrogram i tekst jawny, zaś K jest macierzą o rozmiarze 3×3 reprezentującą klucz szyfru. Jako przykład prześledźmy szyfrowanie tekstu jawnego `paymoremoney` za pomocą klucza

$$\mathrm{K} = \begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix}$$

Sekwencja trzech pierwszych liter tekstu jawnego reprezentowana jest przez wektor wierszowy (15 0 24), mamy zatem

$$(15\ 0\ 24)\mathbf{K} \bmod 26 = (303\ 303\ 531) \bmod 26 = (17\ 17\ 11) = \mathbf{RRL}$$

Powtarzając to postępowanie, otrzymamy kompletny szyfrogram `RRLMWBKA` ↪`SPDH`.

Deszyfracja wykonywana jest za pomocą macierzy odwrotnej do macierzy **K**:

$$\mathbf{P} = \mathbf{CK}^{-1}$$

Obliczamy $\det \mathbf{K} = 23$, $(\det \mathbf{K})^{-1} \bmod 26 = 17$ oraz

$$\mathrm{K}^{-1} = \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix}$$

Istotnie

$$\begin{bmatrix} 17 & 17 & 5 \\ 21 & 18 & 21 \\ 2 & 2 & 19 \end{bmatrix} \begin{bmatrix} 4 & 9 & 15 \\ 15 & 17 & 6 \\ 24 & 0 & 17 \end{bmatrix} = \begin{bmatrix} 443 & 442 & 442 \\ 858 & 495 & 780 \\ 494 & 52 & 365 \end{bmatrix} \bmod 26 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Można łatwo okazać, że zastosowanie macierzy $\mathbf{K}^{-1}$ do szyfrogramu da w rezultacie tekst jawny.

Formalnie rzecz biorąc, szyfrowanie Hilla można zapisać w następującej postaci

$$\mathbf{C} = \mathrm{E}(\mathbf{K}, \mathbf{P}) = \mathbf{PK} \bmod 26$$

$$\mathbf{P} = \mathrm{D}(\mathbf{K}, \mathbf{C}) = \mathbf{PKK}^{-1} \bmod 26 = \mathbf{P}$$

Podobnie jak w przypadku szyfru Playfaira bezpieczeństwo szyfru Hilla wynika stąd, że w szyfrogramie brakuje informacji na temat dystrybucji pojedynczych znaków w tekście jawnym. Efekt maskujący jest tym lepszy, im większy rozmiar ma macierz **K**; użycie macierzy 3×3 maskuje dystrybucję nie tylko pojedynczych znaków, lecz także dystrybucję dwuznaków.

Mimo iż szyfr Hilla potrafi znakomicie przeciwstawić się atakowi ze znanym szyfrogramem, to może być stosunkowo łatwo załamany za pomocą ataku ze znanym tekstem jawnym. Załóżmy, że macierz kluczowa ma rozmiar $m \times m$ i kryptoanalityk dysponuje m parami „tekst jawny – szyfrogram", przy czym w każdej parze tekst jawny i szyfrogram mają długość po m znaków. Oznaczmy każdą parę jako $\langle \mathbf{P}_j, \mathbf{C}_j \rangle$:

$$\mathbf{P}_j = (p_{j1}\, p_{j2} \ldots p_{jm})$$

$$\mathbf{C}_j = (c_{j1}\, c_{j2} \ldots c_{jm})$$

Wówczas

$$\mathbf{C}_j = \mathbf{P}_j\mathbf{K}$$

dla $1 \le j \le m$ i pewnej nieznanej macierzy $\mathbf{K}$. Tworzymy dwie macierze, $\mathbf{X}$ i $\mathbf{Y}$, o rozmiarze $m \times m$, takie że

$$\mathbf{X}_{ij}=p_{ij} \text{ oraz } \mathbf{Y}_{ij}=c_{ij}$$

Prowadzi nas to do równania

$$\mathbf{Y} = \mathbf{XK}$$

Jeśli $\mathbf{X}$ jest macierzą odwracalną, wyliczamy

$$\mathbf{K}=\mathbf{X}^{-1}\mathbf{Y}$$

Jeśli macierz X okaże się być macierzą osobliwą, musimy postarać się o dodatkowe pary „tekst jawny – szyfrogram".

W charakterze przykładu załóżmy, że zaszyfrowaliśmy tekst jawny `hill cipher` za pomocą macierzy 2×2, otrzymując szyfrogram `HCRZSSXNSP`. Mamy więc

$$\begin{bmatrix}7 & 8\end{bmatrix}\mathbf{K} \bmod 26 = \begin{bmatrix}7 & 2\end{bmatrix}$$

$$\begin{bmatrix}11 & 11\end{bmatrix}\mathbf{K} \bmod 26 = \begin{bmatrix}17 & 25\end{bmatrix}$$

i tak dalej. Wykorzystując powyższe dwie pary, otrzymujemy równanie

$$\begin{bmatrix}7 & 2\\ 17 & 25\end{bmatrix} = \begin{bmatrix}7 & 8\\ 11 & 11\end{bmatrix}\mathbf{K} \bmod 26$$

Odwracając macierz $\mathbf{X}$

$$\begin{bmatrix}7 & 8\\ 11 & 11\end{bmatrix}^{-1} = \begin{bmatrix}25 & 22\\ 1 & 23\end{bmatrix}$$

obliczamy macierz kluczową

$$\mathbf{K}=\begin{bmatrix}25 & 22\\ 1 & 23\end{bmatrix}\begin{bmatrix}7 & 2\\ 17 & 25\end{bmatrix} = \begin{bmatrix}549 & 600\\ 398 & 577\end{bmatrix} \bmod 26 = \begin{bmatrix}3 & 2\\ 8 & 5\end{bmatrix}$$

Otrzymany wynik możemy potwierdzić, wykorzystując inne pary „tekst jawny – szyfrogram".

Szyfry polialfabetyczne

Inną metodą zwiększenia bezpieczeństwa szyfru monoalfabetycznego jest użycie wielu podstawień monoalfabetycznych regularnie zmienianych w miarę przetwarzania tekstu jawnego. Szyfry opierające się na tej koncepcji nazywane są ogólnie **szyframi polialfabetycznymi**. Wszystkie one funkcjonują w oparciu o dwie poniższe zasady:

1. Wykorzystywany jest zbiór podstawień monoalfabetycznych.
2. Wybór konkretnego podstawienia dla danej transformacji jest określony przez klucz.

Szyfr Vigenère'a

Najbardziej znanym szyfrem polialfabetycznym — i jednym z najprostszych — jest szyfr Vigenère'a. Jego istotą jest naprzemienne wykorzystywanie podstawień typowych dla szyfru Cezara, ze wszystkimi możliwymi przesunięciami od 0 do 25 (patrz wzór 2.1). Każde z tych podstawień identyfikowane jest przez literę będącą obrazem litery **a** w tym podstawieniu, przykładowo podstawienie z $k = 3$ identyfikowane jest przez literę *d*.

Algorytm szyfrujący Vigenère'a opisać można poglądowo w następujący sposób. Załóżmy, że tekst jawny ma postać $P = p_0\, p_1\, p_2 \ldots p_{n-1}$, zaś klucz ma postać $K = k_0\, k_1\, k_2 \ldots k_{m-1}$, przy czym na ogół $m < n$. Ciąg kolejnych liter szyfrogramu $C = C_0\, C_1\, C_2 \ldots C_{n-1} = \mathrm{E}(K, P) = \mathrm{E}((k_0\, k_1\, k_2 \ldots k_{m-1}), (p_0\, p_1\, p_2 \ldots p_{n-1})) = (p_0 + k_0) \bmod 26 \,||\, (p_1 + k_1) \bmod 26 \,||\, \ldots ||\, (p_{m-1} + k_{m-1}) \bmod 26 \,||\, (p_m + k_0) \bmod 26 \,||\, (p_{m+1} + k_1) \bmod 26 \,||\, \ldots ||\, (p_{2m-1} + k_{m-1}) \bmod 26 \,||\, \ldots$ — i tak dalej (|| oznacza konkatenację ciągów).

Innymi słowy, pierwsza litera tekstu jawnego dodawana jest modulo 26 do pierwszej litery klucza, druga litera tekstu jawnego dodawana jest modulo 26 do drugiej litery klucza i tak dalej, aż *m*-ta litera tekstu jawnego dodana zostanie modulo 26 do *m*-tej litery klucza. Dla następnych liter tekstu jawnego klucz wykorzystywany jest cyklicznie od początku. Postępowanie to kontynuowane jest aż do przetworzenia wszystkich liter tekstu jawnego. Formalnie

$$C_i = (p_i + k_{i \bmod m}) \bmod 26 \qquad \textbf{(2.3)}$$

(porównaj to ze wzorem 2.1 dla szyfru Cezara). Tak więc kolejne znaki tekstu jawnego szyfrowane są przy cyklicznym użyciu kilku szyfrów Cezara, zależnie od kolejnych znaków klucza. Stąd algorytm deszyfrujący również stanowi uogólnienie algorytmu Cezara wyrażonego przez wzór 2.2:

$$p_i = (C_i - k_{i \bmod m}) \bmod 26 \qquad \textbf{(2.4)}$$

Generalnie w przypadku szyfru polialfabetycznego algorytm szyfrujący musi jednoznacznie określać odpowiednią literę klucza dla każdej litery tekstu jawnego,

co można interpretować w ten sposób, że dla teksu jawnego o danej długości dostarczany jest klucz o długości nie mniejszej. Szyfr Vigenère'a realizuje tę zasadę

poprzez cykliczne powtarzanie krótkiego klucza — jeżeli na przykład jest nim słowo *deceptive*, komunikat `we are discovered save yourself` zaszyfrowany zostaje następująco:

```
klucz:          deceptivedeceptivedeceptive
tekst jawny:    wearediscoveredsaveyourself
szyfrogram:     ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

W przełożeniu na numeryczne odpowiedniki liter wygląda to tak:

klucz:	*3*	*4*	*2*	*4*	*15*	*19*	*8*	*21*	*4*	*3*	*4*	*2*	*4*	*15*
tekst jawny:	22	4	0	17	4	3	8	18	2	14	21	4	17	4
szyfrogram:	25	8	2	21	19	22	16	13	6	17	25	6	21	19

klucz:	*19*	*8*	*21*	*4*	*3*	*4*	*2*	*4*	*15*	*19*	*8*	*21*	*4*
tekst jawny:	3	18	0	21	4	24	14	20	17	18	4	11	5
szyfrogram:	22	0	21	25	7	2	16	24	6	11	12	6	9

Ze względu na zmieniające się nieustannie litery klucza dystrybucja liter w tekście otwartym zostaje w szyfrogramie kompletnie zamaskowana, i to niewątpliwie stanowi siłę opisywanego szyfru. Nie oznacza to jednak, że maskowana jest wszelka informacja dotycząca statystyki tekstu jawnego. Na rysunku 2.6 widoczny jest wykres odzwierciedlający dystrybucję znaków w szyfrogramie Vigenère'a utworzonym przy użyciu dziewięcioznakowego klucza; dystrybucja jest tu bardziej „płaska" niż w przypadku szyfru Playfaira, niemniej jednak w szyfrogramie wciąż obecna jest znacząca ilość informacji związanej ze strukturę tekstu jawnego.

Pouczające będzie w tym miejscu zaprezentowanie zarysu metody łamania szyfru Vigenère'a, ponieważ metoda ta odkrywa pewne interesujące zasady matematyczne przydatne w kryptoanalizie.

Załóżmy więc wpierw, iż kryptoanalityk wie, że tekst jawny zaszyfrowany został albo za pomocą podstawienia monoalfabetycznego, albo za pomocą szyfru Vigenère'a. Rozstrzygnięcie tego dylematu nie stanowi dla kryptoanalityka większego problemu: w przypadku szyfru monoalfabetycznego dystrybucja znaków szyfrogramu zbliżona jest do dystrybucji typowej dla konkretnego języka, pokazanej na rysunku 2.5 — w szyfrogramie powinna więc występować jedna litera z częstością zbliżoną do 12,7%, jedna z częstością zbliżoną do 9,06% itp. Oczywiście w przypadku skąpego zasobu materiałów (krótkiego szyfrogramu) trudno oczekiwać dokładnego dopasowania, jeżeli jednak wyraźna jest opisana tendencja, można podejrzewać szyfrowanie monoalfabetyczne.

Jeżeli jednak kryptoanalityk stwierdzi, że ma do czynienia z szyfrem Vigenère'a, dalszy postęp kryptoanalizy uwarunkowany jest (jak za chwilę zobaczymy) trafnym określeniem długości klucza. Niezwykle istotne w tym dziele okazuje się następujące spostrzeżenie: jeżeli dwie identyczne sekwencje znaków występują

w tekście jawnym w odległości[13] będącej wielokrotnością długości klucza, to ich obrazy w szyfrogramie będą identyczne. W naszym przykładzie dwie sekwencje `red` występują w odległości dziewięciu znaków, co wobec również dziewięcioznakowego klucza powoduje, że obie przekształcane są na sekwencje `VTW`; w obu zatem przypadkach do szyfrowania litery `r` wykorzystywana jest litera *e* klucza, do szyfrowania litery `e` — litera *p* klucza, zaś do szyfrowania litery `e` — litera *t* klucza. W rezultacie sekwencja `red` szyfrowana jest jako `VTW` (co zaznaczyliśmy przez podkreślenie liter w szyfrogramie i wyróżnienie komórek tabelki).

Kryptoanalityk dysponujący wyłącznie szyfrogramem zauważy zapewne powtarzające się sekwencje `VTW` w odległości dziewięciu pozycji. Stanowić to będzie dla niego wskazówkę, iż jeżeli nie jest to zbieżność przypadkowa, lecz obrazy dwóch identycznych sekwencji tekstu jawnego, to klucz musi mieć długość 3 albo 9 (liczba 9 nie ma innych dzielników większych niż 1). W dostatecznie długim szyfrogramie takich nieprzypadkowych zbieżności może być stosunkowo dużo, wystarczy więc rozważyć wspólne dzielniki dystansów, o jakie odległe są wystąpienia identycznych sekwencji, bo dzielniki te są prawdopodobnymi długościami klucza.

Kolejny pomyślny krok kryptoanalizy opiera się na kolejnym ważnym spostrzeżeniu: jeżeli długość klucza wynosi *m*, to szyfr stanowi cykliczną kombinację *m* podstawień monoalfabetycznych. W przypadku klucza *deceptive* litery tekstu jawnego występujące na pozycjach 1, 10, 19, 28, 37 itd. szyfrowane są przy użyciu tego samego podstawienia monoalfabetycznego; analiza częstości występowania poszczególnych liter *na tych pozycjach* szyfrogramu odzwierciedla więc dystrybucję ich pierwowzorów w tekście jawnym, identycznie jak w przypadku zwykłego szyfru monoalfabetycznego. Łamanie szyfru Vigenère'a z *m*-znakowym kluczem sprowadza się więc do niezależnego łamania *m* szyfrów monoalfabetycznych.

Opisaną regularność, będącą prostą konsekwencją periodycznego używania tego samego klucza, można wyeliminować, rezygnując ze wspomnianej periodyczności na rzecz klucza stanowiącego konkatenację krótkiego słowa kluczowego i tekstu jawnego będącego przedmiotem szyfrowania. Ten wynalazek, również pochodzący od Vigenère'a, nazywany **systemem autoklucza**, zastosowany do naszego przypadku daje wynik następujący:

```
klucz:         deceptivewearediscoveredsav
tekst jawny:   wearediscoveredsaveyourself
szyfrogram:     ZICVTWQNGKZEIIGASXSTSLVVWLA
```

Nawet jednak i ta komplikacja nie eliminuje podatności szyfru na kryptoanalizę. Jako że klucz i tekst jawny posiadają teraz niemal identyczną dystrybucję liter, użyteczne okazują się techniki statystyczne, przykładowo `e` szyfrowane przez *e* daje w szyfrogramie znak występujący z częstotliwością $(0{,}127)^2 \approx 0{,}016$, `t` szyfrowane

[13] Przez odległość między dwiema sekwencjami rozumiemy dystans dzielący pierwsze znaki tych sekwencji — *przyp. tłum.*

przez t — znak występujący z częstotliwością $(0{,}09)^2 \approx 0{,}008$ (patrz rysunek 2.5). Tego rodzaju regularności mogą doprowadzić do skutecznego odgadnięcia całego tekstu jawnego[14].

Szyfr Vernama

Całkowitą odporność na opisane zabiegi kryptoanalityczne może zapewnić klucz tak samo długi, jak tekst jawny i nie posiadający z tekstem jawnym żadnych związków statystycznych. System szyfrowania oparty na tej zasadzie zaproponował w roku 1918 inżynier z firmy AT & T Gilbert Vernam. System ten operuje na pojedynczych bitach tekstu jawnego i klucza, a wykonywane przezeń przekształcenie można zwięźle zapisać jako

$$c_i = p_i \oplus k_i$$

gdzie

p_i jest i-tym bitem tekstu jawnego

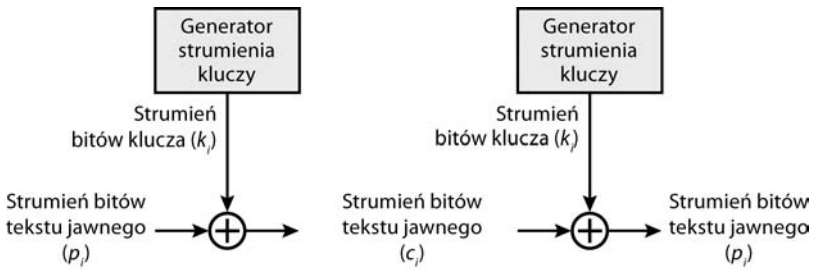
k_i jest i-tym bitem klucza

c_i jest i-tym bitem szyfrogramu

$\oplus$ jest funkcją bitowej różnicy symetrycznej definiowaną następująco:

x	y	x ⊕ y
0	0	0
0	1	1
1	0	1
1	1	0

(Zwróćmy uwagę na podobieństwo tej formuły do wzoru (2.3) określającego szyfrowanie metodą Vigenère'a). Schemat szyfrowania metodą Vernama przedstawiony jest na rysunku 2.7.



Rysunek 2.7. Szyfrowanie metodą Vernama

[14] Mimo iż technologia łamania szyfru Vigenère'a nie jest jakoś specjalnie skomplikowana, jedno z wydań „Scientific American" w 1917 roku zachwalało go jako „impossible of translation". Warto o tym pamiętać, czytając podobne zapewnienia odnoszące się do obecnie konstruowanych systemów kryptograficznych.

Najbardziej bodaj interesującym elementem opisywanego schematu jest generowanie długich kluczy. Vernam zaproponował użycie w tym celu taśmy sklejonej w pętlę, za pomocą której generowano by zbiór cyklicznie powtarzających się kluczy (długich, lecz jednak powtarzających się). Owa cykliczność jest jednak tym elementem, który daje kryptoanalitykowi szansę w przypadku dysponowania przez niego wystarczająco długim szyfrogramem, a jeszcze lepiej odpowiadającą mu porcją tekstu jawnego.

Szyfr z kluczami jednorazowymi

W 1917 roku major armii USA Joseph Mauborgne zaproponował ulepszenie szyfru Vernama, skutkujące absolutną niemożnością jego przełamania: klucz o długości nie mniejszej niż tekst jawny powinien być generowany w sposób losowy oddzielnie dla każdego komunikatu i po wykorzystaniu do zaszyfrowania tego komunikatu nigdy więcej nie używany. System ten, nazywany **szyfrowaniem z kluczami jednorazowymi** (*one-time pad*) eliminuje jakiekolwiek konotacje o charakterze statystycznym między tekstem jawnym a odpowiadającym mu szyfrogramem — szyfrogram ma postać losowego ciągu znaków i jako taki skutecznie opiera się kryptoanalizie.

Zjawisko to stanie się bardziej zrozumiałe po przeanalizowaniu poniższego przykładu. Załóżmy, że używamy zmodyfikowanego szyfru Vigenère'a z 27 znakami, gdzie dodatkowy, 27. znak jest spacją, wykorzystując wyłącznie jednorazowe klucze dla poszczególnych komunikatów. Właśnie udało nam się przechwycić następujący szyfrogram:

```
ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
```

Wiedząc, że mamy do czynienia z szyfrem Vigenère'a, i używając dwóch różnych kluczy, uzyskujemy dwie różne kandydatury na potencjalną postać tekstu jawnego:

```
szyfrogram:   ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
klucz:        pxlmvmsydofuyrvzwc tnlebnecvgdupahfzzlmnyih
tekst jawny:  mr mustard with the candlestick in the hall

szyfrogram:   ANKYODKYUREPFJBYOJDSPLREYIUNOFDOIUERFPLUYTS
klucz:        mfugpmiydgaxgoufhkllmhsqdqogtewbqfgyovuhwt
tekst jawny:  miss scarlet with the knife in the library
```

Który z dwóch jednakowo prawdopodobnych wyników jest tym rzeczywistym? Na to pytanie moglibyśmy próbować odpowiedzieć, dysponując kilkoma innymi szyfrogramami utworzonymi przy użyciu tego samego klucza. Niestety, tym razem klucze są jednorazowe…

Mając dowolny tekst jawny i dowolny szyfrogram o tej samej długości, zawsze możemy znaleźć klucz transformujący ten tekst na rzeczony szyfrogram. Przeszukanie przestrzeni wszystkich możliwych kluczy da intruzowi tylko tyle, że zostanie zasypany lawiną sensownych i prawdopodobnych tekstów jawnych, bez jakiejkolwiek wskazówki co do tego, który z nich jest tym właściwym.

To wszystko wynika z losowego wyboru klucza: szyfrowanie dowolnego tekstu jawnego przy użyciu takiego klucza da w efekcie szyfrogram o losowej statystyce, pozbawiony jakichkolwiek regularności, które mogłyby stanowić punkt wyjścia do kryptoanalizy.

Wspaniale! Mamy upragnione bezpieczeństwo, ale jednocześnie stajemy przed dwoma podstawowymi problemami dotyczącymi samych kluczy, a konkretnie z ich:

1. Generowaniem — intensywnie wykorzystywany system kryptograficzny wymagać może miliona nowych kluczy w ciągu każdej sekundy. Generowanie w takim tempie wysoce losowych wartości jest niebagatelnym wyzwaniem.
2. Dystrybuowaniem — jeszcze poważniejszą jest kwestia przesyłania generowanych „mamucich" kluczy uczestnikom komunikacji, stawiającego wyzwania zarówno pod względem wydajności, jak i pod względem bezpieczeństwa.

Z powyższych względów szyfrowanie z kluczami jednorazowymi użyteczne jest głównie w odniesieniu do informacji, dla której poufność ma znaczenie krytyczne, przesyłanej z niezbyt dużym natężeniem.

Nie zmienia to faktu, że szyfrowanie z kluczami jednorazowymi jest jedynym systemem spełniającym warunki *poufności doskonałej* (*perfect secrecy*) — koncepcję tę wyjaśniamy w dodatku F.

2.3. TECHNIKI PRZESTAWIENIOWE

Wszystkie opisywane dotychczas techniki sprowadzały się do zastępowania symboli tekstu jawnego symbolami szyfrogramu. Odmienną techniką transformowania tekstu jawnego jest zmienianie kolejności (permutowanie) jego symboli. Szyfry opierające się na tej zasadzie nazywane są **szyframi przestawieniowymi**.

Jednym z najprostszych szyfrów tego typu jest **szyfr zygzakowy** (*rail fence*), polegający na zapisywaniu kolejnych symboli wzdłuż przekątnych i odczytywaniu ich wierszami. Przykładowo: szyfrowanie tekstu jawnego `meetmeafterthe ↪togaparty` zygzakiem o głębokości 2 wygląda następująco:

```
m e m a t r h t g p r y
 e t e f e t e o a a t
```

Otrzymaliśmy zatem szyfrogram

```
MEMATRHTGPRYETEFETEOAAT
```

Oczywiście tego rodzaju szyfry okazują się banalne z punktu widzenia kryptoanalizy. Technika nieco bardziej zaawansowana polega na zapisaniu tekstu jawnego w prostokątnej macierzy wierszami, zmianie kolejności kolumn i odczytaniu zawartości macierzy kolejnymi kolumnami — sposób przestawienia (permutacja) kolumn jest właśnie kluczem szyfru. Przykładowo:

```
Klucz:         4 3 1 2 5 6 7
Tekst jawny:   a t t a c k p
               o s t p o n e
               d u n t i l t
               w o a m x y z
Szyfrogram:    TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Klucz ma tutaj postać *4312567*. Aby utworzyć szyfrogram, rozpoczynamy od kolumny etykietowanej numerem 1 i następnie odczytujemy kolumny etykietowane kolejnymi numerami.

Jest oczywiste, że szyfr opierający się wyłącznie na przestawieniach zachowuje statystykę tekstu jawnego, czyli dystrybucję pojedynczych znaków, dwuznaków i trójznaków. W przypadku transpozycji kolumnowej wystarczy wpisać treść szyfrogramu do macierzy i następnie eksperymentować z przestawianiem jej kolumn.

Szyfr przestawieniowy stanie się bardziej bezpieczny, jeśli opierać się będzie na permutacji nieco bardziej skomplikowanej niż transpozycja kolumnowa, na przykład złożeniu dwóch takich (identycznych) transpozycji. Gdy ponownie zaszyfrujemy powyższy szyfrogram tym samym kluczem, otrzymamy:

```
Klucz:     4 3 1 2 5 6 7
Wejście:   t t n a a p t
           m t s u o a o
           d w c o i x k
           n l y p e t z
Wyjście:   NSCYAUOPTTWLTMDNAOIEPAXTTOKZ
```

Aby lepiej uwidocznić wynik owej podwójnej transpozycji, przypiszmy kolejnym literom oryginalnego tekstu jawnego liczby oznaczające ich pozycję w tym tekście jawnym. Dla liczącego 28 znaków tekstu otrzymamy oczywiście

```
01 02 03 04 05 06 07 08 09 10 11 12 13 14
15 16 17 18 19 20 21 22 23 24 25 26 27 28
```

W wyniku pierwszej transpozycji dostaniemy

```
03 10 17 24 04 11 18 25 02 09 16 23 01 08
15 22 05 12 19 26 06 13 20 27 07 14 21 28
```

czyli strukturę dość regularną, ale już po drugiej transpozycji

```
17 09 05 27 24 16 12 07 10 02 22 20 03 25
15 13 04 23 19 14 11 01 26 21 18 08 06 28
```

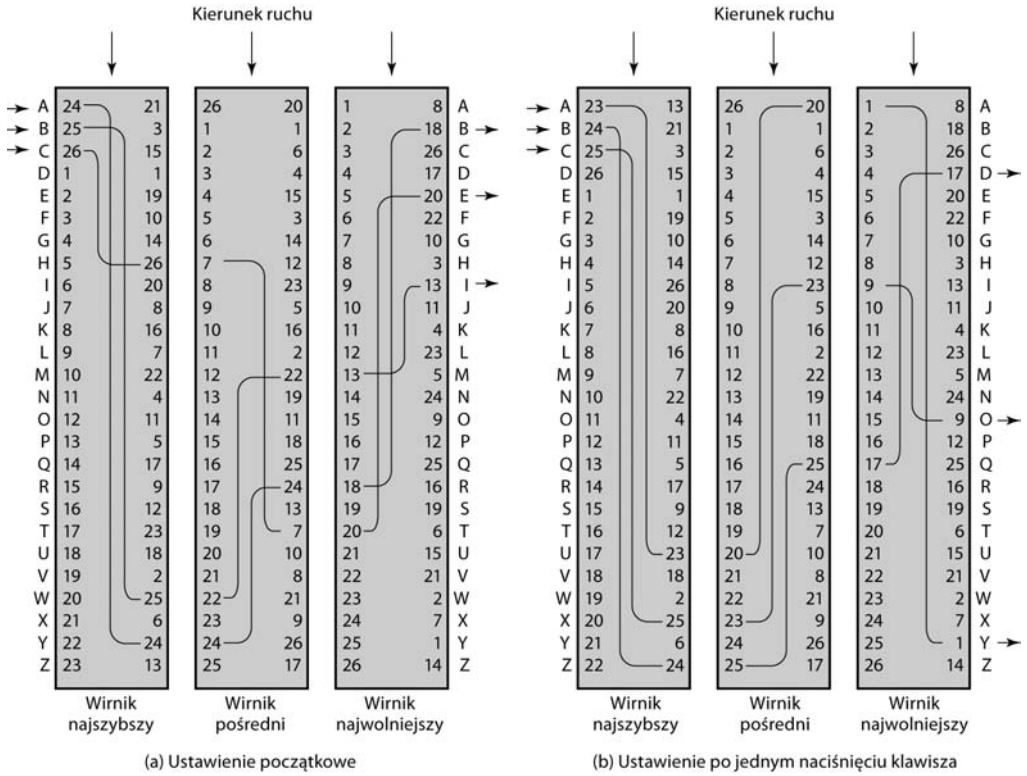
permutacja jest zdecydowanie mniej strukturalna i trudniejsza w kryptoanalizie.

2.4. MASZYNY WIRNIKOWE

Zaprezentowane dotychczas przykłady wyraźnie pokazują, że łączenie wielu etapów szyfrowania prowadzi do algorytmów szyfrujących skutecznie opierających się kryptoanalizie — i to zarówno w przypadku szyfrów podstawieniowych, jak

i przestawieniowych. Zanim wynaleziono szyfr DES, idea ta realizowana była powszechnie przez klasę urządzeń, które ogólnie nazwać można **maszynami wirnikowymi** (*rotor machines*)[15].

Podstawowa koncepcja maszyny wirnikowej przedstawiona jest na rysunku 2.8. Maszyna ta składa się z kilku cylindrów obracających się niezależnie, choć sprzężonych w specyficzny sposób (o tym za chwilę). Każdy cylinder realizuje połączenie 26 styków wejściowych z 26 stykami wyjściowymi za pomocą układu przewodów; na rysunku dla przejrzystości pokazano jedynie trzy połączenia w każdym wirniku.



Rysunek 2.8. Schemat maszyny wirnikowej z trzema cylindrami

W części (a) rysunku styk skojarzony z literą A skojarzony jest z parą styków nr 24 pierwszego cylindra; gdy operator naciśnie na klawiaturze literę A, impuls elektryczny doprowadzony zostanie przewodami do przedostatniego styku wyjściowego.

[15] Maszyny funkcjonujące na bazie połączonych wirników używane były w czasie II wojny światowej przez Niemcy (Enigma) i Japonię (Purple). Złamanie obu szyfrów przez aliantów i polskich kryptologów (przede wszystkim Mariana Rejewskiego, Jerzego Różyckiego i Henryka Zygalskiego — już w 1932 roku!) miało decydujący wpływ na przebieg i wynik wojny.

Każde kolejne naciśnięcie klawisza spowoduje obrót cylindra o jedną pozycję (w konwencji rysunku 2.8 będzie to przesunięcie z góry na dół), co prowadzi do sytuacji przedstawionej w części (b) rysunku: naciśnięcie litery A w tej sytuacji prowadzić będzie impuls do ostatniego styku.

Rzecz jasna każda pozycja cylindra odpowiada pewnemu szyfrowaniu monoalfabetycznemu, za pomocą pojedynczego cylindra możemy więc zrealizować szyfr polialfabetyczny złożony o okresie 26. To oczywiście rozwiązanie banalne z perspektywy kryptoanalizy i maszyna posiadająca tylko jeden wirnik nie na wiele by się zdała. Opisana zasada zyskuje praktyczne znaczenie dopiero wtedy, gdy maszynę wyposażymy w kilka wirników połączonych ze sobą w sposób kaskadowy — czyli taki, że kompletny obrót danego cylindra powoduje przesunięcie cylindra następnego o jedną pozycję (podobnie jak np. w samochodowym liczniku kilometrów). Szyfr polialfabetyczny realizowany przez maszynę o n wirnikach jest więc szyfrem o okresie 26^n, dla $n = 3$ daje to $26 \times 26 \times 26 = 17\,576$. Dodanie czwartego i piątego cylindra zwiększa tę wartość do (odpowiednio) 456 976 i 11 881 376. Jak to trafnie i obrazowo ujął David Kahn w swej książce [KAHN96] na stronie 413:

> Szyfr o takim okresie czyni daremnymi wszelkie próby jego złamania w oparciu o częstość występowania liter. Każda taka próba wymagałaby co najmniej 50 liter dla każdego monoalfabetu, a to oznacza, że układ wirników musiałby wykonać 50 razy swój pełny cykl. Otrzymany szyfrogram byłby bardziej obszerny niż zapis wszystkich wystąpień w Senacie i Izbie Reprezentantów w trakcie trzech kolejnych sesji Kongresu. Żaden kryptoanalityk nie podejmie się zadania, na wykonanie którego po prostu nie wystarczyłoby mu całego życia; nawet dyplomaci, którzy potrafią być równie gadatliwi jak politycy, rzadko kiedy wspinają się na takie wyżyny słowotoku.

Z perspektywy dzisiejszej kryptologii znaczenie maszyn wirnikowych jest o tyle istotne, że utorowały one drogę do jednego z najczęściej używanych dziś szyfrów — DES. Piszemy o nim w rozdziale 3.

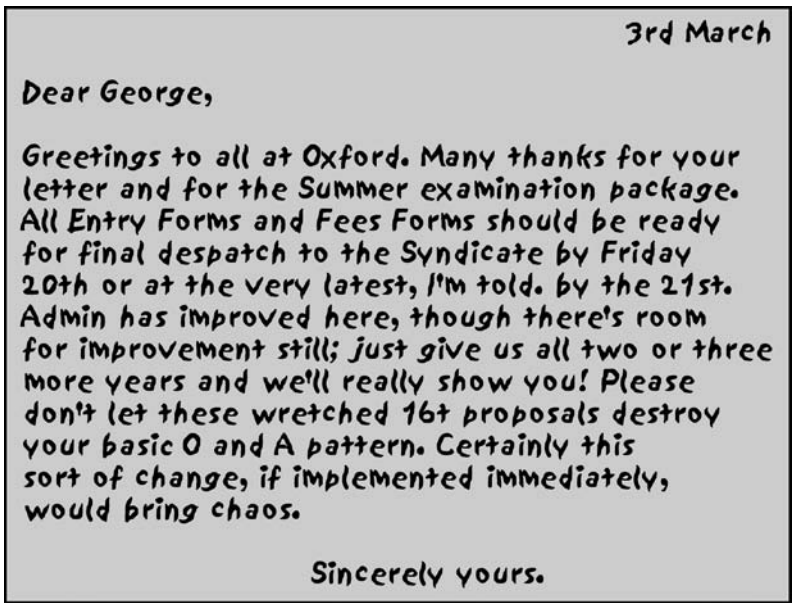
2.5. STEGANOGRAFIA

Zakończymy ten rozdział omówieniem techniki, której tak naprawdę nie da się zaliczyć do kryptografii, ale która podobnie jak kryptografia służy ukrywaniu informacji.

Podczas gdy kryptografia wiąże się z dokonywaniem transformacji czyniących przesyłany komunikat nieczytelnym dla outsiderów, metoda zwana **steganografią** zmierza do ukrycia samego faktu istnienia komunikatu[16].

[16] Korzenie steganografii sięgają V wieku przez Chrystusem. W ciągu wieków przyjmowała ona różne formy, by wreszcie zyskać nowy wymiar w dzisiejszych realiach — o czym obszernie pisze David Kahn w cytowanej już książce [KAHN96].

Najprostszą, choć czasochłonną w konstruowaniu formą zastosowania stenografii jest użycie pozornie banalnego układu słów lub liter, skrywającego jednak istotny komunikat, na przykład uformowany z pierwszych liter wspomnianych słów. Przykład takiego układu pokazano na rysunku 2.9 — podzbiór widocznych słów składa się na istotny komunikat, co czytelnicy sami mogą potwierdzić przy odrobinie wysiłku[17].



Rysunek 2.9. Puzzle inspektora Morse z kryminału Colina Dextera „Świat ciszy Nicholasa Quinna"

[17] Dla bardziej niecierpliwych czytelników rozwiązanie:

Drogi George,

Pozdrowienia dla wszystkich w Oxfordzie. Dzięki za **Twoje**
listy, dla sesji letniej egzaminacyjnej **przesyłki**
formularzy wszystkich podań i wpłat **gotowe**
do ostatecznego nadania do Konsorcjum w **piątek**
20., no, powiedzmy nie później niż **21**.
Szef zatwierdził zadania, jeszcze dziś opuszczą **pokój**
dziekana; daj nam jeszcze dwa czy **trzy**
lata, to dopiero im pokażemy! **Proszę**,
nie daj tej nieszczęsnej pozycji 16+ **zniszczyć**
a zwłaszcza wzorców O i A, bo **to**
się chyba jeszcze zmieni; wprowadzając je **natychmiast**,
moglibyśmy spowodować zamieszanie...

Gwoli wyjaśnienia ewentualnych wątpliwości: by zachować treść ukrytego komunikatu, zmieniłem nieco tekst ze względu na słowo *room* występujące w oryginale, odnoszące się zarówno do dokumentów (*room for improvement* — „miejsce na poprawki"), jak i do pomieszczenia w budynku (słowo „pokój" jest tu istotne) i brak analogicznego słowa w języku polskim — *przyp. tłum.*

W przeszłości steganografia przyjmowała jeszcze inne formy, o czym pisze L. Myers w swej książce [MYER91], wymieniając między innymi:

- **znakowanie liter** — wybrane litery rękopisu lub maszynopisu „poprawiane" były ołówkiem, co jednak widoczne było tylko przy oglądaniu dokumentu pod odpowiednim kątem w stosunku do padającego światła;
- **atrament sympatyczny** — substancja pełniąca rolę atramentu pozostawała niewidoczna bez poddania jej odpowiedniej obróbce fizykochemicznej;
- **perforowanie dokumentu** — dyskretna perforacja na wybranych literach widoczna była jedynie podczas oglądania dokumentu „pod światło";
- **niedostrzegalne interlinie** — między oficjalnymi, czarnymi wierszami maszynopisu znajdowały się wiersze „wystukane" przy użyciu taśmy korekcyjnej, widoczne jedynie przy silnym oświetleniu.

Wymienione techniki, cokolwiek archaiczne, posiadają jednak współczesne odpowiedniki. P. Wayner w książce [WAYN93] opisuje wykorzystanie najmniej znaczących bitów bitmapy fotografii jako nośnika ukrytego komunikatu. Przykładowo: format Photo CD Kodaka zapewnia maksymalną rozdzielczość 2048×3072 piksele przy kodowaniu koloru każdego piksela na 24 bitach, po 8 bitów na każdą składową RGB. Wykorzystując najmniej znaczące bity w każdej ze wspomnianych składowych, zyskujemy przestrzeń do zakodowania 18 megabitów, czyli 2,25 megabajta informacji bez zauważalnego zniekształcenia kolorystyki fotografii. Ponadto jeżeli „wbudowywana" w ten sposób informacja nie jest oryginalnym komunikatem, lecz wynikiem jego skompresowania, to po pierwsze: oryginalny komunikat może być jeszcze większy, po drugie: osoba podejrzewająca jego istnienie nie będzie miała praktycznie żadnej możliwości zweryfikowania swych przypuszczeń. Dostępnych jest wiele gotowych narzędzia automatyzujących opisany proces.

Steganografia charakteryzuje się kilkoma niedogodnościami, które obce są kryptografii. Przede wszystkim uderzający jest duży narzut „szumu" towarzyszącego użytecznej informacji — w opisanej powyżej metodzie użyteczny jest tylko jeden z ośmiu bitów, pozostałe pełnią rolę maskującą. Jeżeli ponadto fakt (i sposób) przemycania informacji w bitmapach zostanie wykryty, staje się praktycznie bezużyteczny, choć można temu przeciwdziałać, stosując opisane kompresowanie lub uprzednie szyfrowanie komunikatu.

Na szczęście jest i zaleta. Steganografia może być stosowana przez partnerów, którzy mogliby wiele stracić w przypadku wykrycia samego faktu ich komunikowania się (niekoniecznie wykrycia treści komunikacji). Natrafienie na szyfrowaną komunikację jednoznacznie identyfikuje jej uczestników jako osoby zdecydowanie mające coś do ukrycia.

2.6. ZALECANE MATERIAŁY UZUPEŁNIAJĄCE

Wszystkim zainteresowanym tworzeniem i łamaniem kodów można polecić książkę Davida Kahna [KAHN96]. Mimo iż jej treść koncentruje się raczej na przejawach obecności kryptologii niż jej aspektach technicznych, stanowi znakomite wprowadzenie w temat i czyta się ją z zainteresowaniem. Inną wspaniałą książką traktującą kryptologię w ujęciu historycznym jest [SING99].

Skondensowane ujęcie technik opisywanych w tym rozdziale (i kilka innych rzeczy) znajdą czytelnicy w książce [GARD72]. Istnieje wiele książek traktujących klasyczną kryptografię z bardziej technicznego punktu widzenia. Jedną z najbardziej godnych polecenia wydaje się [SINK66]. Zachwycająca książka [KORN96] zawiera obszerne omówienie klasycznych technik szyfrowania. Dwie książki dostarczające dużej ilości technicznych szczegółów tych technik to [GARR01] i [NICH99]. Czytelnicy szczególnie zainteresowani tematem z pewnością docenią dwutomowe dzieło R. Nicholsa [NICH96] szczegółowo opisujące szyfrowanie konwencjonalne, wraz z zestawem testowych szyfrogramów, z gotowymi rozwiązaniami.

Opis maszyn wirnikowych wraz z omówieniem problematyki łamania produkowanych przez nie szyfrogramów dostępny jest w książce I. Kumara [KUMA97].

W książce [KATZ00] S. Katzenbeisser wyczerpująco omawia steganografię; innym dobrym źródłem wiedzy z tej dziedziny jest książka P. Waynera [WAYN96].

GARD72 Gardner M., *Codes, Ciphers, and Secret Writing*. New York, Dover, 1972.

GARR01 Garrett P., *Making, Breaking Codes: An Introduction to Cryptology*. Upper Saddle River, NJ, Prentice Hall, 2001.

KAHN96 Kahn D., *The Codebreakers: The Story of Secret Writing*. New York, Scribner, 1996.

KATZ00 Katzenbeisser S. (red.), *Information Hiding Techniques for Steganography and Digital Watermarking*. Boston:Artech House, 2000.

KORN96 Korner T., *The Pleasures of Counting*. Cambridge, England, Cambridge University Press, 1996.

KUMA97 Kumar I., *Cryptology*. Laguna Hills, CA, Aegean Park Press, 1997.

MYER91 Myers L., *Spycomm: Covert Communication Techniques of the Underground*. Boulder, CO: Paladin Press, 1991.

NICH96 R. Nichols *Classical Cryptography Course*. Laguna Hills, CA, Aegean Park Press, 1996.

NICH99 R. Nichols (red.) *ICSA Guide to Cryptography*. New York, McGraw-Hill, 1999.

SING99 S. Singh *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. New York: Anchor Books, 1999.

SINK66 A. Sinkov *Elementary Cryptanalysis: A Mathematical Approach*. Washington, D.C., The Mathematical Association of America, 1966.

WAYN96 P. Wayner *Disappearing Cryptography*. Boston, AP Professional Books, 1996.

Polecane strony WWW

- **American Cryptogram Association**: strona stowarzyszenia kryptografów amatorów. Zawiera wiele informacji o szyfrowaniu i linki do stron poświęconych klasycznej kryptografii.
- **Crypto Corner**: strona Simona Singha. Zawiera mnóstwo wartościowej informacji oraz interaktywne narzędzia wspomagające naukę kryptografii.
- **Steganography**: bogata kolekcja dokumentów i linków do stron o tej tematyce.

2.7. KLUCZOWE TERMINY, PYTANIA PRZEGLĄDOWE I PROBLEMY

Kluczowe terminy

Algorytm bezpieczny bezwarunkowo (*unconditionally secure algorithm*)
Algorytm bezpieczny obliczeniowo (*computationally secure algorithm*)
Atak siłowy (*brute-force attack*)
Deszyfracja (*deciphering*, *decryption*)
Dwuznak (*digram*)
Klucz jednorazowy (*one-time pad*)
Kryptoanaliza (*cryptanalysis*)
Kryptografia (*cryptography*)
Kryptologia (*cryptology*)
Steganografia (*steganography*)
System kryptograficzny (*cryptographic system*)
Szyfr (*cipher*)
Szyfr blokowy (*block cipher*)
Szyfr Cezara (*Caesar cipher*)
Szyfr Hilla (*Hill cipher*)
Szyfr monoalfabetyczny (*monoalphabetic cipher*)
Szyfr Playfaira (*Playfair cipher*)
Szyfr podstawieniowy (*substitution cipher*)
Szyfr polialfabetyczny (*polyalphabetic cipher*)
Szyfr przestawieniowy (*transposition cipher*)
Szyfr strumieniowy (*stream cipher*)
Szyfr Vernama (*Vernam cipher*)
Szyfr Vigenère'a (*Vigenère cipher*)
Szyfr zygzakowy (*rail fence cipher*)
Szyfrogram (*ciphertext*)
Szyfrowanie (*enciphering, encryption*)
Szyfrowanie konwencjonalne (*conventional encryption*)
Szyfrowanie symetryczne (*symmetric encryption*)
Szyfrowanie z pojedynczym kluczem (*single-key encryption*)
Tekst jawny (*plaintext*)

Pytania przeglądowe

2.1. Jakie są podstawowe elementy szyfru symetrycznego?
2.2. Jakie dwie podstawowe operacje wykonywane są przez algorytm szyfrujący?
2.3. Ilu kluczy potrzebują uczestnicy komunikujący się za pomocą szyfrowanej informacji?
2.4. Jaka jest różnica między szyfrem blokowym a szyfrem strumieniowym?
2.5. Jakie są dwa zasadnicze podejścia do łamania szyfrów?
2.6. Wymień i krótko zdefiniuj typy ataków kryptoanalitycznych wyróżniane w oparciu o informację dostępną dla kryptoanalityka.
2.7. Jaka jest różnica między algorytmem bezpiecznym bezwarunkowo a algorytmem bezpiecznym obliczeniowo?
2.8. Opisz krótko założenia szyfru Cezara.
2.9. Zdefiniuj pojęcie szyfru monoalfabetycznego.
2.10. Opisz krótko założenia szyfru Playfaira.
2.11. Jaka jest różnica między szyfrem monoalfabetycznym a szyfrem polialfabetycznym?
2.12. Jakie dwa podstawowe problemy związane są z szyfrowaniem w oparciu o klucze jednorazowe?
2.13. Co to jest szyfr przestawieniowy?
2.14. Co to jest steganografia?

Problemy

2.1. Uogólnienie szyfru Cezara, znane jako *afiniczny szyfr Cezara*, opiera się na następującym algorytmie szyfrującym, przekształcającym literę tekstu jawnego p na literę szyfrogramu C:

$$C = \mathrm{E}([a, b], p) = (ap + b) \bmod 26$$

Podstawowym wymaganiem pod adresem każdego algorytmu szyfrującego jest różnowartościowość realizowanego przezeń przekształcenia, czyli wymaganie spełnienia warunku

$$\text{jeśli } p \neq q, \text{ to } \mathrm{E}(k, p) \neq \mathrm{E}(k, q)$$

w przeciwnym razie niemożliwa będzie deszyfracja szyfrogramu, gdyż określonej jego literze odpowiadać będzie kilka różnych liter tekstu jawnego. Afiniczny szyfr Cezara nie spełnia podanego warunku, gdyż na przykład gdy $a = 2$ i $b = 3$, to $\mathrm{E}([a, b], 0) = \mathrm{E}([a, b], 13) = 3$.

a) Czy możliwe jest wymuszenie spełnienia wspomnianego warunku przez narzucenie określonych ograniczeń na wartość b? Dlaczego tak lub dlaczego nie?

b) Jakie wartości niedozwolone są dla parametru a?

c) Przedstaw i uzasadnij ogólne formuły na zbiory wartości dozwolonych i wartości niedozwolonych dla parametru a.

2.2. Ile istnieje afinicznych szyfrów Cezara spełniających warunek różnowartościowości?

2.3. W szyfrogramie utworzonym za pomocą szyfru afinicznego najczęściej występującą literą jest B, na drugim miejscu w rankingu częstości występowania plasuje się litera U. Zaproponuj sposób łamania tego szyfru.

2.4. Poniższy szyfrogram utworzony został przez prosty algorytm podstawieniowy:

```
53‡‡†305))6*;4826)4‡.)4‡);806*;48†8¶60))85;;]8*;:‡*8†83
(88)5*†;46(;88*96*?;8)*‡(;485);5*†2:*‡(;4956*2(5*—4)8¶8*
;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†528806*81
(‡9;48;(88;4(‡?34;48)4‡;161;:188;‡?;
```

Odtwórz tekst jawny w języku angielskim.

Wskazówki:

1. Jak wiadomo, najczęściej występującą literą w tekstach w języku angielskim jest `e`, zatem pierwszy lub drugi (być może trzeci?) znak w rankingu częstotliwości występowania w szyfrogramie jest obrazem litery `e`. Ponadto litera `e` ma tendencję częstego występowania parami (`meet`, `fleet`, `speed`, `seen`, `been`, `agree` itp.). Rozpocznij więc od znalezienia obrazu litery `e`.
2. Najczęściej występującym w języku angielskim słowem jest `the`. Wykorzystaj ten fakt do znalezienia obrazów liter `t` i `h`.
3. Odszyfruj resztę szyfrogramu przez dedukcję pozostałych słów.

Uwaga: Zaszyfrowany komunikat jest poprawnym tekstem w języku angielskim, lecz niekoniecznie musi wyglądać sensownie przy pierwszym czytaniu.

2.5. Jednym ze sposobów rozwiązania problemu dystrybucji kluczy jest wykorzystywanie w tym celu określonego wiersza (linii) z książki, którą posiadają nadawca i odbiorca; zwykle (przynajmniej w szpiegowskich kryminałach) kluczem jest pierwsze zdanie z książki. Wykorzystywany tu schemat pochodzi z powieści Ruth Rendell *Talking to Strange Men*; spróbuj jednak rozwiązać go samodzielnie, bez zaglądania do treści książki.

Rozpatrzmy następujący szyfrogram:

```
SIDKHKDM AF HCRKIABIE SHIMC KD LFEAILA
```

Został on utworzony na podstawie klucza stanowiącego pierwsze zdanie powieści *The Other Side of Silence* (opisującej historię szpiega Kima Philby'ego):

```
The snow lay thick on the steps and the snowflakes driven
by the wind looked black in the headlights of the cars.
```

Wykorzystano prosty szyfr podstawieniowy.

a) Jak wygląda algorytm szyfrujący?
b) Jak bezpieczny jest ten algorytm?
c) Aby maksymalnie uprościć problem uzgadniania klucza, uczestnicy komunikacji wybierają pierwsze albo ostatnie zdanie książki jako klucz. Zdecydowanie częściej wybierane jest pierwsze zdanie — dlaczego?

2.6. W jednej z prowadzonych spraw Sherlock Holmes stanął przed problemem rozszyfrowania następującego komunikatu:

```
534 C2 13 127 36 31 4 17 21 41
DOUGLAS 109 293 5 37 BIRLSTONE
26 BIRLSTONE 9 127 171
```

Mimo zakłopotania Watsona Holmes natychmiast wydedukował typ szyfru. Czy Ty też potrafisz?

2.7. Poniższy problem jest autentyczny, pochodzi ze starego, obecnie dostępnego publicznie, podręcznika U.S. Special Forces (kopia dostępna jest na stronie WWW związanej z niniejszą książką).

a) Wykorzystując dwa klucze: *`cryptografic`* i *`network security`*, zaszyfruj następujący komunikat:

```
Be at the third pillar from the left outside the lyceum
theatre tonight at seven. If you are distrustful bring
two friends.
```

(komunikat ten pochodzi z noweli o Sherlocku Holmesie *The Sign of Four*). Przyjmij rozsądne założenia odnośnie do powtarzających się liter kluczy, nadmiarowych liter w kluczach oraz traktowania spacji i znaków przestankowych w tekście jawnym.

b) Rozszyfruj szyfrogram, pokazując wykonywane operacje krok po kroku.

c) Jakie Twoim zdaniem są zalety opisanej techniki i kiedy opłaca się ją stosować?

2.8. Podstawową niedogodnością związaną z szyframi monoalfabetycznymi jest konieczność zapisywania do pamięci permutowanych sekwencji szyfrogramu. Podstawowym sposobem unikania tej niedogodności jest użycie słowa kluczowego umożliwiającego dynamiczne generowanie tych sekwencji. Przykładowo: jeżeli słowem tym jest *CIPHER*, konkatenujemy z nim ciąg ułożonych alfabetycznie liter nie występujących w nim, przyporządkowując kolejne znaki całości kolejnym znakom tekstu jawnego:

```
tekst jawny: a b c d e f g h i j k l m n o p q r s t u v w x y z
szyfrogram:  C I P H E R A B D F G J K L M N O Q S T U V W X Y Z
```

Dla bezpieczeństwa można dodatkowo zapisać całość w układzie wierszowym macierzy i następnie odczytywać ją kolumnami:

```
C I P H E R
A B D F G J
K L M N O Q
S T U V W X
Y Z
```

W efekcie otrzymamy sekwencję

```
C A K S Y I B L T Z P D M U H F N V E G O W R J Q X
```

Tę właśnie technikę zastosowaliśmy do zaszyfrowania komunikatu `it was disclosed yesterday...` w sekcji 2.2 — znajdź użyte słowo kluczowe.

2.9. Gdy kuter torpedowy PT-109, dowodzony przez porucznika J.F. Kennedy'ego, został zatopiony przez japoński niszczyciel, w australijskiej stacji odbiorczej odebrano następujący szyfrogram utworzony przy użyciu szyfru Playfaira:

```
KXJEY UREBE ZWEHE WRYTU HEYFS
KREHE GOYFI WTTTU OLKSY CAJPO
BOTEI ZONTX BYBNT GONEY CUZWR
GDSON SXBOU YWRHE BAAHY USEDQ
```

Znajdź tekst jawny, wiedząc, że użyty klucz miał postać *royal new zealand navy*. Dwuznak `TT` w szyfrogramie jest obrazem dwuznaku `tt` w tekście jawnym.

2.10. **a)** Skonstruuj macierz Payfaira dla słowa kluczowego *largest*.

b) Skonstruuj macierz Payfaira dla słowa kluczowego *occurrence*; przyjmij rozsądne założenie dotyczące traktowania zdublowanych liter.

2.11. **a)** Używając poniższej macierzy Playfaira

M	F	H	I/J	K
U	N	O	P	Q
Z	V	W	X	Y
E	L	A	R	G
D	S	T	B	C

zaszyfruj następujący komunikat (zaczerpnięty z historii o Sherlocku Holmesie *The Adventure of the Bruce-Partington Plans*):

```
Must see you over Cadogan West. Coming at once.
```

a) Powtórz ćwiczenie a), używając macierzy Playfaira z problemu 2.10 a.
b) Jak oceniasz bezpieczeństwo uzyskanych szyfrogramów? Czy potrafisz uogólnić swe spostrzeżenia?

2.12. Ile potencjalnie możliwych kluczy można użyć na potrzeby szyfru Playfaira? Zignoruj fakt, że niektóre klucze mogą dawać takie same szyfrogramy. Wyraź swoją odpowiedź w postaci przybliżenia będącego potęgą liczby 2.

2.13. Jaki system szyfru podstawieniowego uzyskamy, posługując się macierzą Playfaira o rozmiarach 25×1?

2.14. **a)** Zaszyfruj komunikat `meet me at the usual place at ten rather than eight oclock` za pomocą macierzy Hilla $\begin{pmatrix} 9 & 4 \\ 5 & 7 \end{pmatrix}$. Pokaż szczegóły i wynik obliczeń.
b) Pokaż szczegóły obliczeń związanych z deszyfracją otrzymanego szyfrogramu.

2.15. Jak pokazaliśmy, szyfr Hilla jest nieodporny na atak ze znanym tekstem jawnym, jeśli kryptoanalityk dysponuje wystarczającą liczbą par „tekst jawny – szyfrogram". Jeszcze prościej można złamać szyfr Hilla, przypuszczając atak z wybranym tekstem jawnym — opisz szczegóły takiego ataku.

2.16. Można pokazać, że macierz $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ używana w algorytmie szyfrującym Hilla musi spełniać następujący warunek: jej wyznacznik ($ad - bc$) musi być względnie pierwszy z liczbą 26, czyli nie może dzielić się przez 2, 13 ani 26. Określ liczbę różnych macierzy Hilla 2×2 spełniających ten warunek, bez kolejnego wyliczania ich wszystkich, lecz przez zastosowanie poniższego rozumowania:
a) Określ liczbę macierzy, których wyznacznik jest parzysty dlatego, że parzyste są oba wiersze (wiersz uważamy za „parzysty", jeśli parzyste są oba jego elementy).
b) Określ liczbę macierzy, których wyznacznik jest parzysty dlatego, że parzyste są obie kolumny (kolumnę uważamy za „parzystą", jeśli parzyste są oba jej elementy).
c) Określ liczbę macierzy, których wyznacznik jest parzysty dlatego, że wszystkie elementy są nieparzyste.
d) Określ liczbę macierzy, których wyznacznik jest parzysty z innych powodów.
e) Określ liczbę macierzy, których wyznacznik jest podzielny przez 13, ponieważ elementy pierwszej kolumny są podzielne przez 13.
f) Określ liczbę macierzy, których pierwsza kolumna nie jest podzielna przez 13, lecz wyznacznik jest podzielny przez 13, ponieważ druga kolumna jest wielokrotnością pierwszej modulo 13.
g) Określ liczbę wszystkich macierzy, których wyznacznik jest podzielny przez 13.
h) Określ liczbę wszystkich macierzy, których wyznacznik jest wielokrotnością 26 z powodu spełnienia pary warunków (a, e), (b, e), (c, e), (a, f) itp.
i) Określ liczbę wszystkich macierzy, których wyznacznik jest nieparzysty i niepodzielny przez 13.

2.17. Używając szyfru Vigenère'a, zaszyfruj słowo `explanation` za pomocą klucza *`leg`*.

2.18. Ten problem związany jest z używaniem kluczy jednorazowych w szyfrowaniu Vigenère'a. Klucz generowany jest jako strumień liczb pseudolosowych z przedziału

od 0 do 26; każda z liczb określa przesunięcie podobne jak w przypadku szyfru Cezara, tak więc na przykład wobec sekwencji *3 19 5* … pierwsza litera tekstu jawnego przesuwana jest cyklicznie o 3 pozycje, druga — o 19 pozycji, trzecia — o 5 pozycji itd.

a) Zaszyfruj tekst jawny `sendmoremoney`, używając klucza

9 0 1 7 23 15 21 14 11 11 2 8 9

b) Wykorzystując szyfrogram otrzymany w ćwiczeniu a), znajdź klucz przekształcający go w tekst jawny `cashnotneeded`.

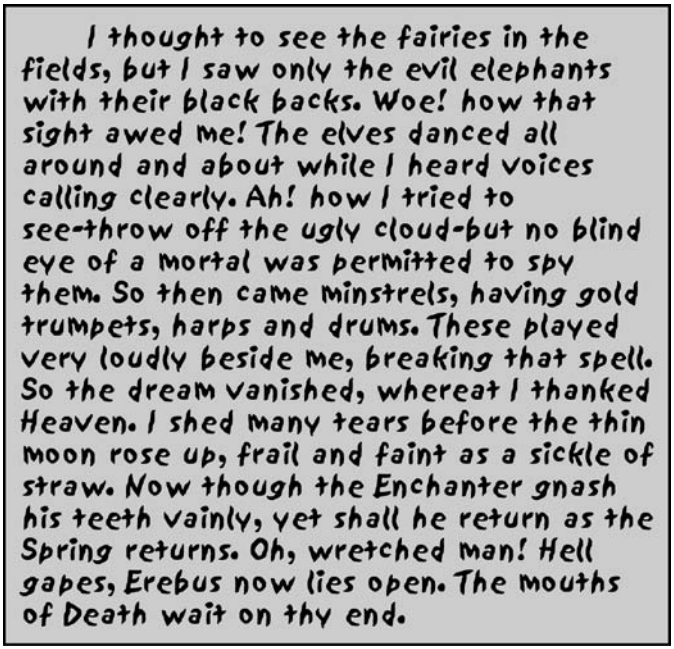
2.19. W jednej z powieści Dorothy Sayers lord Peter staje przed zadaniem zinterpretowania komunikatu widocznego na rysunku 2.10. Odkrywa on jednocześnie klucz do tej interpretacji, będący następującą sekwencją liczb całkowitych:

```
78765654343211234345656787887876565 4
34321123434565678788787656544332112 34
```

a) Rozszyfruj komunikat. *Wskazówka*: jaka jest największa spośród wymienionych liczb całkowitych?

b) Jak bezpieczny jest ten szyfr, gdy znany jest algorytm, lecz nieznany jest klucz?

c) Jak bezpieczny jest ten szyfr, gdy algorytm nie jest znany, lecz znany jest klucz?



Rysunek 2.10. Układanka lorda Petera

Zadania programistyczne

2.20. Napisz program realizujący uogólnione szyfrowanie Cezara, znane także jako szyfr addytywny.

2.21. Napisz program realizujący afiniczne szyfrowanie Cezara, opisane w problemie 2.1.

2.22. Napisz program wykonujący (bez udziału człowieka) atak na addytywny szyfr Cezara w oparciu o częstość występowania liter. Program powinien produkować propozycje tekstów jawnych, określając w przybliżeniu prawdopodobieństwo każdej z nich. Byłoby dobrze, gdyby program mógł realizować polecenia w rodzaju „Podaj 10 najbardziej prawdopodobnych tekstów jawnych".

2.23. Napisz program wykonujący (bez udziału człowieka) atak na dowolny monoalfabetyczny szyfr podstawieniowy w oparciu o częstość występowania liter. Program powinien produkować propozycje tekstów jawnych, określając w przybliżeniu prawdopodobieństwo każdej z nich. Byłoby dobrze, gdyby program mógł realizować polecenia w rodzaju „Podaj 10 najbardziej prawdopodobnych tekstów jawnych".

2.24. Napisz program wykonujący szyfrowanie i deszyfrację w oparciu o macierze Hilla rozmiaru 2×2.

2.25. Napisz program dokonujący efektywnego ataku ze znanym tekstem jawnym na szyfr Hilla o znanym wymiarze macierzy (m). Jaka jest złożoność czasowa tego algorytmu w funkcji m?

SKOROWIDZ

A

B

C

D

E

F

G

H

I

J

K

L

Ł

M

N

O

P

R

S

T

U

V

W

X

Z

Ź

Wirusy, hakerzy, szpiegostwo gospodarcze, elektroniczne podsłuchy i kradzieże — era internetu ma także swoją ciemną stronę, która stawia przed nami coraz większe wyzwania w zakresie bezpieczeństwa informacji. Dla większości organizacji kwestie ochrony dostępu do danych przechowywanych w systemach komputerowych i wymienianych między nimi, zachowania tajności wiadomości oraz skutecznego odpierania ataków stały się zagadnieniami kluczowymi, mogącymi przesądzić o ich istnieniu. Bezpieczeństwo sieci ma także ogromne znaczenie dla zwykłych użytkowników internetu, często przetrzymujących na dyskach ważne, poufne dokumenty i dokonujących za pomocą sieci rozmaitych transakcji. Na szczęście dziś mamy już świetnie przetestowane, dojrzałe technologie i narzędzia, które dają nam naprawdę ogromne możliwości w zakresie ochrony i szyfrowania danych. Jedyne, czego Ci trzeba, to wiedza, jak je skutecznie wykorzystać.

Oto pierwszy z dwóch tomów kompletnego przewodnika po praktycznych zastosowaniach kryptografii i innych mechanizmów bezpieczeństwa w celu ochrony informacji i sieci. Ten adresowany zarówno do studentów, jak i zawodowców podręcznik podzielono na trzy naszpikowane wiedzą i ciekawymi przykładami części, wprowadzające kolejno w szyfry symetryczne, szyfry asymetryczne i kryptograficzne algorytmy ochrony integralności danych. Przeczytasz tu m.in. na temat trybów operacyjnych szyfrów blokowych oraz przyjrzysz się standardowi AES i generowaniu liczb pseudolosowych. Otrzymasz obszerną prezentację algorytmów kryptograficznych i doskonały przewodnik po metodach uwierzytelniania. Ponadto nauczysz się efektywnie wykorzystywać system Sage — wieloplatformowe, darmowe narzędzie z użytecznym, elastycznym i łatwym do opanowania systemem obliczeń algebraicznych związanych z kryptografią. Znajdziesz tu także gotowe dla tego systemu przykłady, ilustrujące praktyczne zastosowania teorii liczb i algorytmów kryptograficznych.

William Stallings *jest autorem siedemnastu książek z zakresu technicznych aspektów bezpieczeństwa informacji i sieci komputerowych. Jest jedenastokrotnym laureatem nagrody za najlepszą książkę informatyczną roku, przyznawanej przez Text and Academic Authors Association. W trakcie ponadtrzydziestoletniej kariery zawodowej zaprojektował i zaimplementował wiele pakietów związanych z protokołami TCP/IP i OSI dla różnych platform. Jako konsultant doradzał m.in. agencjom rządowym oraz dostawcom sprzętu i oprogramowania.*

- Ogólny zarys bezpieczeństwa komputerowego
- Szyfrowanie symetryczne
- Klasyczne techniki szyfrowania
- Szyfry blokowe i standard DES
- Podstawy teorii liczb i ciał skończonych
- Standard AES
- Tryby operacyjne szyfrów blokowych
- Generatory liczb pseudolosowych i szyfry strumieniowe
- Szyfrowanie asymetryczne
- Wstęp do teorii liczb
- Kryptografia z kluczami publicznymi i szyfr RSA
- Inne systemy kryptografii z kluczami publicznymi
- Kryptograficzne algorytmy ochrony integralności danych
- Kryptograficzne funkcje haszujące
- Uwierzytelnianie komunikatów
- Podpisy cyfrowe

helion.pl
księgarnia internetowa

Nr katalogowy: 66537

Księgarnia internetowa:
http://helion.pl

Zamówienia telefoniczne:
0 801 339900
0 601 339900

Helion

Sprawdź najnowsze promocje:
http://helion.pl/promocje
Książki najchętniej czytane:
http://helion.pl/bestsellery
Zamów informacje o nowościach:
http://helion.pl/nowosci

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel.: 32 230 98 63
e-mail: helion@helion.pl
http://helion.pl

Informatyka w najlepszym wydaniu

Cena 99,00 zł

ISBN 978-83-246-2986-2