ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1997
Seria: INFORMATYKA z. 32 Nr kol. 1356

Katarzyna TRYBICKA, Adam ZIĘBIŃSKI
Politechnika Śląska, Instytut Informatyki

PROGRAMOWE I SPRZĘTOWE IMPLEMENTACJE ALGORYTMÓW KRYPTOGRAFICZNYCH DLA SIECI KOMPUTEROWYCH

Streszczenie. Publikacja jest zapowiedzią pracy na temat systemów kryptograficznych w informatyce, a w szczególności opracowania programowego szyfratora pracującego w systemie prywatnym wg algorytmu IDEA.

HARDWARE AND SOFTWARE SOLUTIONS OF CRYPTOGRAPHIC SYSTEMS

Summary. The publication is a communique on the subject of cryptographic systems, particulary cryptographic algorithms which work like the private key systems. The attention to the IDEA encryption scheme is paid later in this publication.

1. Systemy kryptograficzne

W dniu dzisiejszym już nikogo nie trzeba przekonywać o konieczności szyfrowania informacji. Rozwój systemów kryptograficznych, a szczególnie ich masowe zastosowanie do magazynowania i transmisji informacji w takich m.in. dziedzinach, jak bankowość, handel czy przemysł, uwypukliły problem ochrony zasobów banków danych oraz konieczność zabezpieczania procesów technologicznych.

Jednym z elementów ochrony są środki techniczne, wśród których są również zabezpieczenia kryptograficzne. Przed współczesnymi systemami kryptograficznymi stawia się przede wszystkim zadania ochrony danych przechowywanych w systemach komputerowych lub przesyłanych liniami telekomunikacyjnymi, a stopień ich bezpieczeństwa uzależniony jest od zabezpieczenia przed ujawnieniem, modyfikacją lub zniszczeniem.

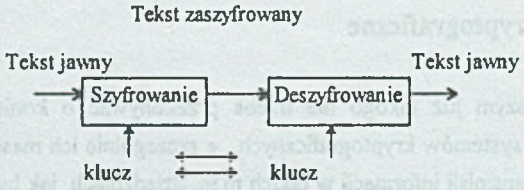
Wyszczególnia się następujące kryteria, jakie musi spełniać algorytm kryptograficzny:

- ma zapewniać wysoki stopień bezpieczeństwa,
- powinien być kompletnie określony i łatwy do zrozumienia,
- bezpieczeństwo musi zależeć tylko i wyłącznie od klucza,
- algorytm powinien być dostępny dla wszystkich użytkowników,
- powinien mieć zdolność adaptacji do różnych zastosowań,
- musi dawać możliwość ekonomicznej, elektronicznej implementacji,
- ma być efektywny w użyciu,
- musi dawać możliwość sprawdzania jego poprawności,
- powinien spełniać warunki jego eksportu.

1.1. Dostępność klucza

W zależności od sposobu dostępności kluczy szyfrujących i deszyfrujących można wyróżnić (wg [1]) dwa typy systemów kryptograficznych:

- systemy prywatne, które wyróżnia fakt, że klucz szyfrujący jest zarazem kluczem deszyfrującym. Podstawy teoretyczne systemów prywatnych zostały opracowane w końcu lat czterdziestych tego stulecia przez C.E. Shanona i rozwinięte później przez M.E. Hallmana [2][3];
- systemy publiczne, które charakteryzują się tym, że klucz szyfrujący i klucz deszyfrujący są różne oraz że klucz szyfrujący jest ogólnie dostępny, natomiast klucz deszyfrujący posiadają wyłącznie osoby upoważnione. Wynalezienie systemów publicznych datuje się na połowę lat siedemdziesiątych XX wieku.



Rys. 1. Prywatny system kryptograficzny
Fig. 1. The private key system

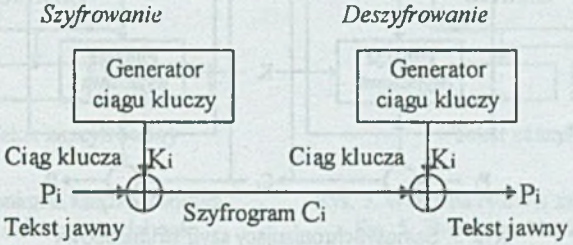
1.2. Algorytmy

Ze względu na sposób, w jaki algorytmy przekształcają tekst jawny w szyfrogram, można wyróżnić algorytmy typu strumieniowego i algorytmy blokowe.

1.2.1. Szyfry strumieniowe

Tekst jawny jest szyfrowany kolejno bit po bicie (rys. 2.).

Jak nietrudno zauważyć, bezpieczeństwo całkowicie zależy od generatora strumienia kluczy. Można sobie wyobrazić sytuację, w której generator strumienia klucza wytwarza nieskończenie wiele zer, co sprawi, że na wyjściu otrzymamy szyfrogram równy tekstowi jawnemu. Może się też zdarzyć, że generator będzie wytwarzał powtarzający się wzorzec, co w efekcie sprawi, że nasz cały złożony system będzie zwykłym sumatorem modulo dwa. Trzeba by więc stworzyć generator, który wytwarzałby nieskończony strumień bitów losowych (nie pseudolosowych!). W ten sposób otrzymamy klucz jednorazowy i doskonałe zabezpieczenie. W rzeczywistości generatory ciągów klucza wytwarzają strumienie bitów, które są zdeterminowane. Mogą więc one być w stosunkowo łatwy sposób odtworzone przez kryptoanalityków. Jedynym rozwiązaniem wydaje się stworzenie generatora, który wytwarzałby losowo wyglądający ciąg bitów, zadanie to jednak nie należy do łatwych.



Rys. 2. Generator strumienia klucza
Fig. 2. The keystream generator

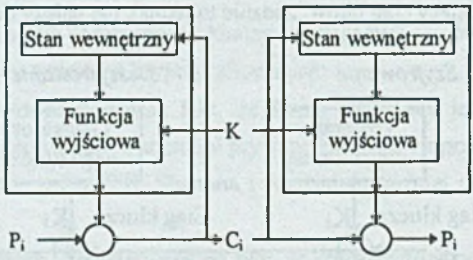
W kategorii szyfrów strumieniowych można wyróżnić synchroniczne szyfry strumieniowe oraz samosynchronizujące szyfry strumieniowe. W pierwszym przypadku mamy do czynienia z sytuacją, w której klucz jest generowany niezależnie od strumienia wiadomości. Po obu stronach, szyfrującej i deszyfrującej, generatory strumieni kluczy wytwarzają identyczne strumienie bitów klucza. Wszystko działa poprawnie do chwili, gdy oba generatory są zsynchronizowane. Może jednak nastąpić „zagubienie" bitu podczas przesyłania, lub jeden z generatorów „przeskoczy" cykl; wówczas od tego momentu każdy znak szyfrogramu będzie odszyfrowywany niepoprawnie. By ową pomyłkę wyeliminować, nadawca i odbiorca muszą ponownie zsynchronizować swoje generatory strumieni kluczy. Nie jest to jednak takie proste, jakby się na pozór wydawało; muszą dokonać tego w taki sposób, aby nie powtórzyła się żadna część ciągu klucza. Nie wystarczy więc ponowna inicjacja obu generatorów.

Pozytywną stroną tego rozwiązania jest to, iż synchroniczne szyfry strumieniowe nie rozsiewają błędów transmisji.

Szyfry strumieniowe pracują w dwu trybach:

- tryb sprzężenia zwrotnego wyjściowego - gdzie klucz wpływa na funkcję następnego stanu; funkcja wyjścia jest prosta i nie zależy od klucza,
- tryb licznikowy - gdzie mamy do czynienia z prostymi funkcjami stanu wewnętrznego i skomplikowanymi funkcjami wyjścia, które są zależne od klucza.

W samosynchronizujących szyfrach strumieniowych każdy bit ciągu szyfrującego jest funkcją pewnej stałej liczby poprzednich bitów szyfrogramu. Najczęściej te szyfry strumieniowe pracują w trybie sprzężenia zwrotnego szyfrogramu. Na rysunku 3 przedstawiony jest schemat działania szyfru strumieniowego w trybie sprzężenia zwrotnego szyfrogramu.



Rys. 3. Samosynchronizujący szyfr strumieniowy
Fig. 3. The self - synchronous stream cipher

Ponieważ stan wewnętrzny zależny jest od *n* poprzednich bitów szyfrogramu, więc generator strumienia klucza do odszyfrowania będzie automatycznie zsynchronizowany z generatorem strumienia klucza do szyfrowania po odebraniu *n* bitów szyfrogramu.

Niestety, ujemną stroną samosynchronizujących się szyfrów strumieniowych jest propagacja błędów. Jeżeli wystąpi błąd jednego bitu w czasie przesyłania, to po stronie deszyfrującej otrzymamy *n* niepoprawnych bitów.

1.2.2. Szyfry blokowe

Algorytmy blokowe działają z reguły na blokach tekstu jawnego lub szyfrogramu długości 64 bitów. Można w ich pracy wyróżnić cztery podstawowe tryby:

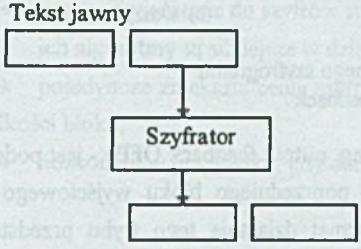
- **Tryb elektronicznej książki kodowej** (ang. electronic codebook ECB) - jest najłatwiejszym i najszybszym trybem do zastosowania w szyfrach blokowych. Działanie ECB przedstawia rysunek 4.

Jego zaletą jest możliwość niezależnego szyfrowania każdego bloku tekstu jawnego. Nie musimy liniowo szyfrować pliku. Ta cecha jest istotna dla plików, w których musi być zastosowana możliwość dostępu losowego, na przykład w bazach danych.

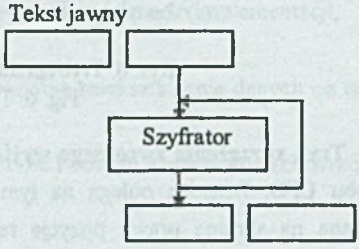
Wadą jest jednak to, iż jest on najsłabszy (najłatwiejszy do złamania). Nie powinno się więc z tego trybu korzystać.

- **Tryb wiązania bloków zaszyfrowanych** (ang. cipher block chaining CBC) - tekst jawny jest przed szyfrowaniem sumowany modulo dwa z poprzednimi blokami szyfrogramów. Działanie CBC przedstawia rysunek 5.

W tym trybie nie można zacząć szyfrowania, zanim nie odbierze się całego bloku danych. Jest nieprzydatny, jeżeli dane muszą być przetwarzane w porcjach wielkości bajtu.



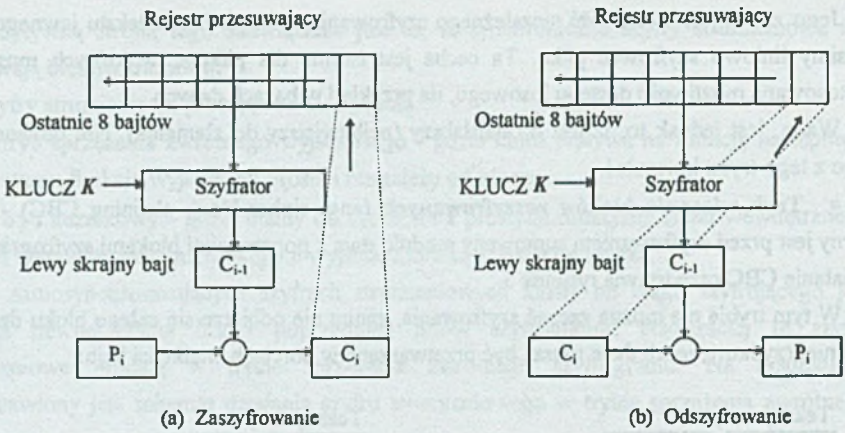
Rys. 4. Elektroniczna książka kodowa
Fig. 4. The electronic codebook



Rys. 5. Wiązania bloków zaszyfrowanych
Fig. 5. The cipher block chaining

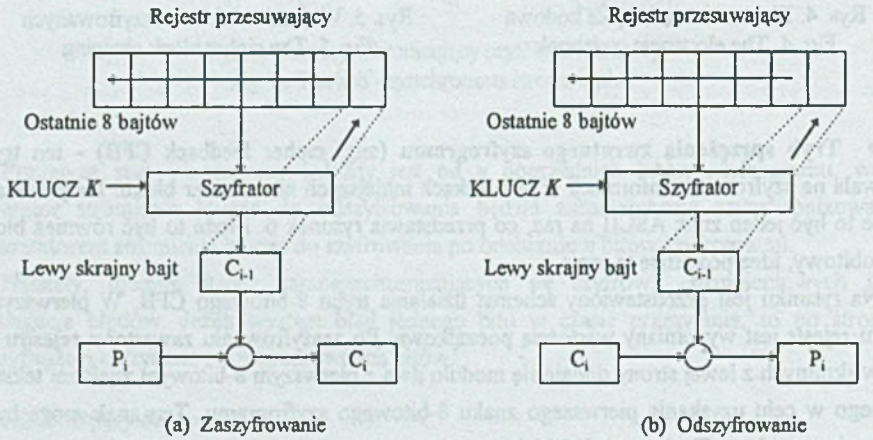
- **Tryb sprzężenia zwrotnego szyfrogramu** (ang. cipher feedback CFB) - ten tryb pozwala na szyfrowanie informacji w jednostkach mniejszych niż rozmiar bloku. Na przykład może to być jeden znak ASCII na raz, co przedstawia rysunek 6. Może to być również blok jednobitowy, idea pozostaje ta sama.

Na rysunku jest przedstawiony schemat działania trybu 8-bitowego CFB. W pierwszym kroku rejestr jest wypełniany wartością początkową. Po zaszyfrowaniu zawartości rejestru 8 bitów skrajnych z lewej strony dodaje się modulo dwa z pierwszym 8-bitowym znakiem tekstu jawnego w celu uzyskania pierwszego znaku 8-bitowego szyfrogramu. Ten znak może być następnie przesłany. Tych samych 8 bitów przesuwa się na 8 bitów skrajnych z prawej strony kolejki, a wszystkie pozostałe bity są przesuwane o 8 pozycji w lewo. Odrzuca się najbardziej znaczących 8 bitów. Następny bit szyfruje się dokładnie w ten sam sposób. Ten tryb pracy jest wolniejszy od swoich poprzedników.



Rys. 6. Tryb sprzężenia zwrotnego szyfrogramu
Fig. 6. The cipher feedback

- **Tryb sprzężenia zwrotnego wyjściowego** (ang. output feedback OFB) - jest podobny do trybu CFB. Różnica polega na tym, że część poprzedniego bloku wyjściowego jest kierowana na skrajne prawe pozycje rejestru. Schemat działania tego trybu przedstawia rysunek 7.



Rys. 7. Tryb sprzężenia zwrotnego wyjściowego
Fig. 7. The output feedback

1.2.3. Porównanie szyfrów strumieniowych z blokowymi

Szyfry strumieniowe:

- szyfrują i deszyfrują po jednym bicie danych na raz, co w praktyce oznacza, że nie są odpowiednie do implementacji programowej (operacje na bitach są czasochłonne),
- algorytmy są łatwe do analizy matematycznej,
- pojedyncze zniekształcenie szyfrogramu powoduje zniekształcenie tylko jednego bitu tekstu jawnego,
- nadaje się do szyfrowania danych typu ASCII z terminala komputerowego; blokowy algorytm szyfrowania w tym wypadku nie spełni stawianych przed nim oczekiwań.

Szyfry blokowe:

- w przeciwieństwie do szyfrów strumieniowych są łatwiejsze do implementacji,
- ich algorytmy są silniejsze w działaniu,
- pojedyncze zniekształcenie szyfrogramu powoduje zniekształcenie danych co najmniej wielkości bloku,
- doskonałe w przypadku, gdy dane są zapisywane i odczytywane w postaci bloków.

2. Algorytmy kryptograficzne

2.1. Standard szyfrowania DES (ang. Data Encryption Standard)

Powstał w latach siedemdziesiątych i został przyjęty jako standard szyfrowania przez Amerykański Narodowy Instytut Standaryzacji (ang. American National Standards Institute - ANSI) 23 listopada 1976 roku.

DES jest szyfrem blokowym, pracującym na 64-bitowych pakietach danych. Zarówno do szyfrowania, jak i deszyfrowania stosuje się ten sam algorytm. Klucz jest 64-bitowy, przy czym informacja użyteczna zajmuje 56 bitów (co ósmy bit w ciągu klucza jest bitem parzystości). Całe bezpieczeństwo spoczywa właśnie na nim.

Algorytm DES to kombinacja dwu podstawowych technik: mieszania i rozpraszania.

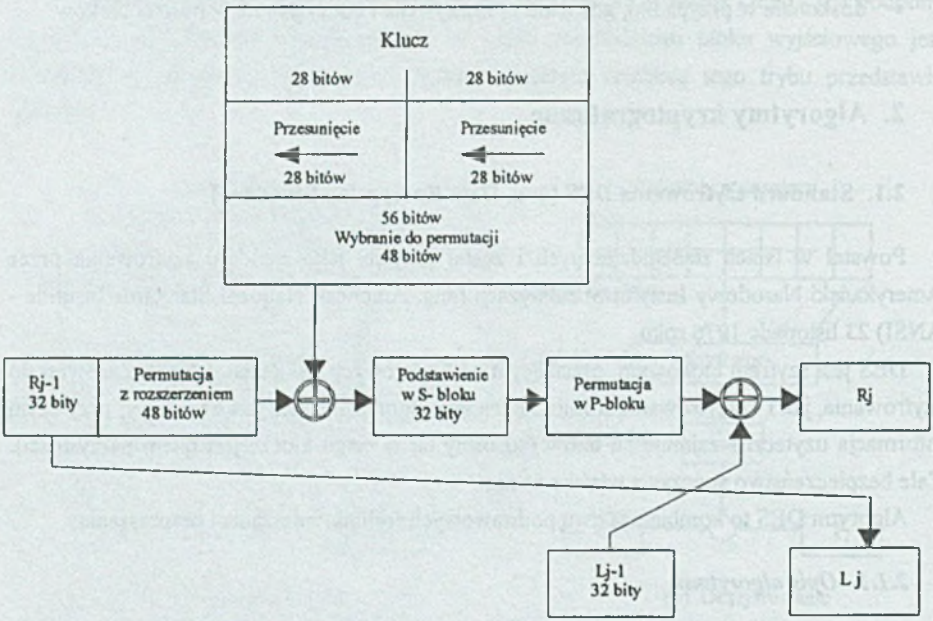
2.1.1. Opis algorytmu

Tekst jawny (64-bitowy blok) poddawany jest permutacji wstępnej (blok oznaczony IP). Potem dzielony jest na dwa podciągi 32-bitowe. Następnie wykonywanych jest 16 cykli jednakowych operacji, nazywanych funkcjami *f*, w czasie których są łączone z kluczem. Po szesnastym cyklu lewa i prawa strona są łączone i poddawane permutacji końcowej.

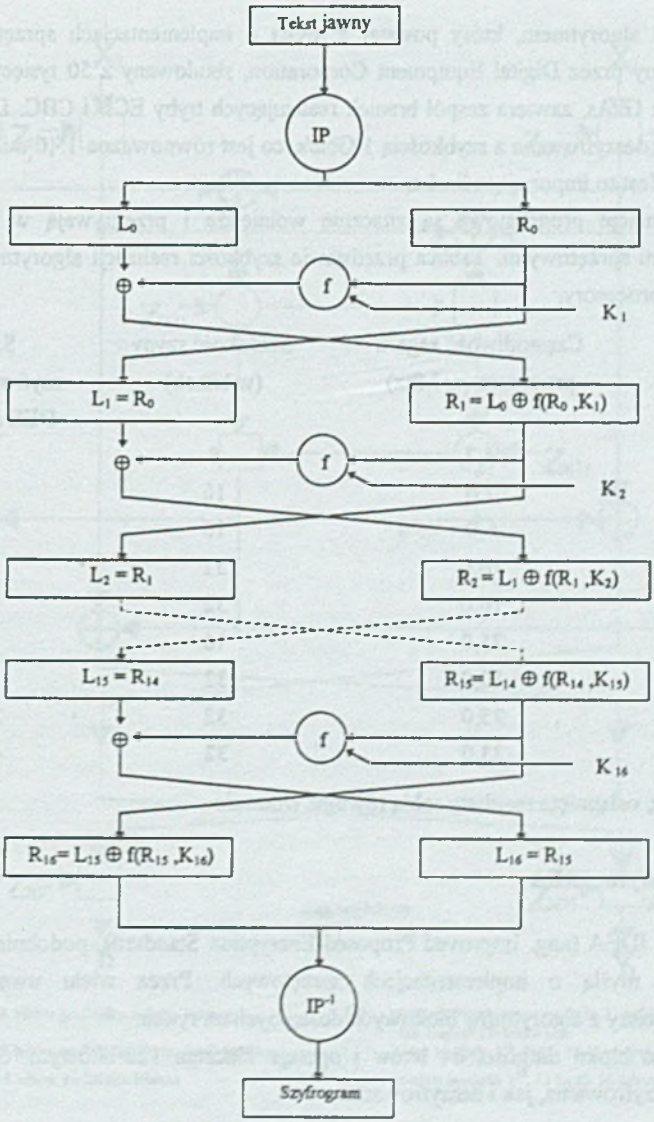
2.1.1.1. Przekształcenia klucza.

Ponieważ klucz jest 64-bitowy, redukowany jest do klucza 56 bitów przez pominięcie co ósmego bitu parzystości. Tak przygotowany ciąg bitów dzielony jest na dwa podciągi 28-bitowe. Następnie połowy te przesuwane są w lewo o jeden lub dwa bity, zależnie od numeru cyklu. Po połączeniu nowo powstałych ciągów wybiera się 48 z 56 bitów (permutacja z kompresją). Tak otrzymujemy klucz dla i-cyklu (gdzie i jest numerem cyklu).

W funkcji *f* prawa połowa bloku danych jest poddawana permutacji z rozszerzeniem, czyli z 32 do 48 bitów. Następnie łączony jest za pomocą poelementowej sumy modulo 2 z 48 bitami przesuniętego i spermutowanego klucza. Po tej operacji otrzymany ciąg dzielony jest na 8 części i wprowadzany do skrzynek S, gdzie z 6-bitowych podciągów na wyjściu otrzymujemy 4-bitowe podciągi, które łączymy ze sobą i otrzymujemy zaszyfrowany ciąg 32-bitowy.



Rys. 8. Metoda wyznaczania wartości funkcji f dla DES
Fig. 8. Methoda AssignnIng of dependent variable of f function for DES algorithm



Rys. 9. Schemat blokowy algorytmu DES
Fig. 9. The block diagram of DES algorithm

2.1.2. Sprzętowe i programowe implementacje DES

DES jest algorytmem, który powstał z myślą o implementacjach sprzętowych. Układ skonstruowany przez Digital Equipment Corporation, zbudowany z 50 tysięcy tranzystorów na podłożu z GaAs, zawiera zespół bramek realizujących tryby ECB i CBC. Dane mogą być szyfrowane i deszyfrowane z szybkością 1 Gbit/s, co jest równoważne 15,6 milionom bloków na sekundę. Jest to imponująca liczba.

Implementacje programowe są znacznie wolniejsze i przegrywają w konkurencji z rozwiązaniami sprzętowymi. Tablica przedstawia szybkości realizacji algorytmu DES przez różne mikroprocesory.

Procesor	Częstotliwość zegara procesora (w Mhz)	Szerokość szyny (w bitach)	Szybkość szyfrowania (bloki DES na sekundę)
8088	4,7	8	370
68000	7,6	16	900
80286	6,0	16	1100
68020	16,0	32	3500
68030	16,0	32	3900
80386	25,0	16	5000
68030	50,0	32	9600
68040	25,0	32	23200
80486	33,0	32	40600

Jak widać, osiągnięte rezultaty robią również wrażenie.

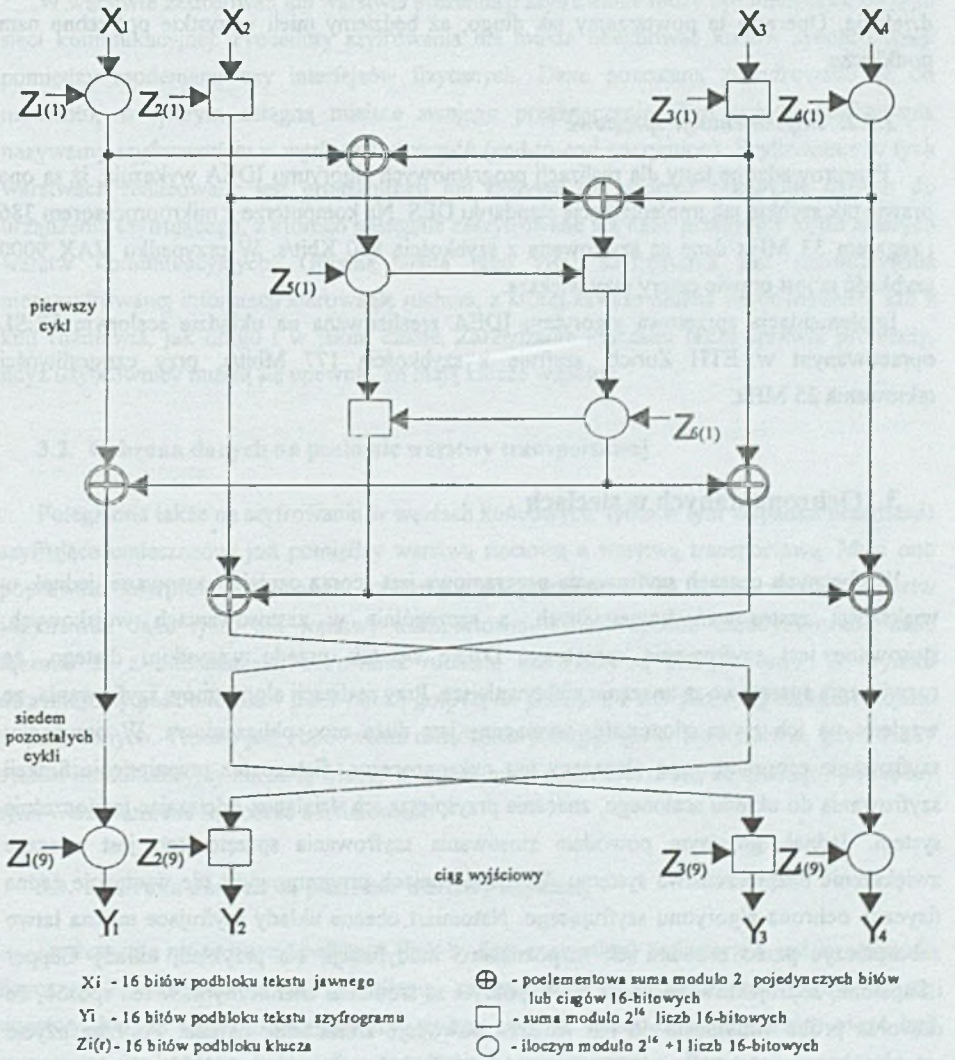
2.2. Algorytm IDEA

Algorytm IDEA (ang. Improved Proposed Encryption Standard), podobnie jak DES, był tworzony z myślą o implementacjach sprzętowych. Przez wielu uważany jest za najbezpieczniejszy z algorytmów blokowych dostępnych na rynku.

Pracuje na bloku długości 64 bitów i operuje kluczem 128-bitowym. Stosowany jest zarówno do szyfrowania, jak i deszyfrowania.

2.2.1. Opis algorytmu

Rysunek 9 przedstawia ogólny szkic algorytmu. Jak widać, blok danych dzielony jest na cztery podbloki (16-bitowe), które stanowią wejście do pierwszego cyklu algorytmu. Cykli wykonywanych jest osiem. Jak nietrudno zauważyć, algorytm wykorzystuje aż 52 podklucze o długości 16-bitów każdy. Uzyskujemy je dzieląc nasz klucz wejściowy na osiem podciągów.



Rys. 10. Schemat blokowy algorytmu DES
Fig. 10. The block diagram of DES algorithm

W rezultacie tego działania otrzymamy osiem pierwszych podkluczy. By uzyskać kolejne, przesuwamy cyklicznie klucz wejściowy o 25 bitów w lewo i ponownie wykonujemy dzielenie. Operacje te powtarzamy tak długo, aż będziemy mieli wszystkie potrzebne nam podklucze.

2.2.2. Implementacje sprzętowe

Przeprowadzone testy dla realizacji programowych algorytmu IDEA wykazują, iż są one prawie tak szybkie jak implementacje standardu DES. Na komputerze z mikroprocesorem 386 i zegarem 33 MHz dane są szyfrowane z szybkością 880 Kbit/s. W przypadku VAX 9000 szybkość ta jest prawie cztery razy większa.

Implementacja sprzętowa algorytmu IDEA zrealizowana na układzie scalonym VLSI, opracowanym w ETH Zurich, szyfruje z szybkością 177 Mbit/s, przy częstotliwości taktowania 25 MHz.

3. Ochrona danych w sieciach

W obecnych czasach szyfrowanie programowe jest coraz częściej stosowane, jednak w większości zastosowań komercyjnych, a szczególnie w zastosowaniach wojskowych, stosowane jest szyfrowanie sprzętowe. Dzieje się tak przede wszystkim dlatego, że rozwiązania sprzętowe są znacznie efektywniejsze. Przy realizacji algorytmów szyfrowania, ze względu na ich sporą złożoność, wymagana jest duża moc obliczeniowa. W przypadku szyfrowania programowego obciążany jest mikroprocesor, dlatego też przeniesienie funkcji szyfrowania do układu scalonego, znacznie przyśpiesza ich działanie, odciążając jednocześnie system. Jednak głównym powodem stosowania szyfrowania sprzętowego jest znaczne zwiększenie bezpieczeństwa systemu. W rozwiązaniach programowych nie występuje żadna fizyczna ochrona algorytmu szyfrującego. Natomiast obecne układy szyfrujące można łatwo zabezpieczyć przed próbami ich rozpoznania i modyfikacji. Na przykład, układy Clipper i Capstone, zaprojektowane przez NSA, pokryte są środkami chemicznymi w ten sposób, że dowolna próba wniknięcia do ich wnętrza powoduje zniszczenie układu. Poprzez użycie promieniowania elektromagnetycznego można niekiedy ujawnić, co dzieje się wewnątrz układów, dlatego wymagane jest odpowiednie ich ekranowanie. Za stosowaniem rozwiązań sprzętowych przemawia również łatwość instalowania, gdyż o wiele prościej jest włożyć sprzęt szyfrujący do modemów, telefonów, faksów, komputerów, niż połączyć go z oprogramowaniem. W przypadku szyfrowania informacji w sieciach komputerowych może być ono realizowane na dowolnej warstwie protokołów modelu komunikacyjnego OSI.

3.1. Ochrona danych na poziomie warstwy prezentacji

W warstwie zastosowań lub warstwie prezentacji szyfrowanie może być niezależne od typu sieci komunikacyjnej. Procedury szyfrowania nie muszą obejmować kodów synchronizacji pomiędzy modemami czy interfejsów fizycznych. Dane pozostaną zaszyfrowane aż do momentu, w którym osiągną miejsce swojego przeznaczenia. Ten sposób szyfrowania nazywamy *szyfrowaniem w węzłach końcowych* (end-to-end encryption). Szyfrowanie w tych warstwach realizowane jest programowo lub odbywa się poprzez przesłanie danych do urządzenia szyfrującego, z którego następnie zaszyfrowane już dane przesyłane są do niższych warstw komunikacyjnych. Główną wadą tego typu szyfrowania jest pozostawienie niezaszyfrowanej informacji kierowania ruchem, z której zawsze można się dowiedzieć, kto z kim rozmawia, jak długo i w jakim czasie. Zarządzanie kluczami także sprawia problemy, gdyż użytkownicy muszą się upewnić, że mają klucze wspólne.

3.2. Ochrona danych na poziomie warstwy transportowej

Polega ona także na szyfrowaniu w węzłach końcowych, tylko w tym wypadku urządzenie szyfrujące umieszczone jest pomiędzy warstwą sieciową a warstwą transportową. Musi ono poprawnie interpretować dane w zależności od protokołów dla trzech dolnych warstw i szyfrować dane tylko dla warstwy transportowej. W ten sposób zmodyfikowane dane łączone są z informacjami sterowania ruchem, które nie są zaszyfrowane. W wyniku otrzymujemy podobne jak w p.3.1 ramkę gotową do przesyłu, z zaszyfrowanymi informacjami w polu danych. Trudne jest zbudowanie urządzenia pracującego w tej warstwie, gdyż każdy system komunikacyjny ma swój własny protokół, przy czym zdarza się, że interfejsy pomiędzy tymi warstwami nie są dobrze zdefiniowane.

3.3. Ochrona danych na poziomie warstwy fizycznej

Szyfrowanie na poziomie połączeń (link-by-link encryption) realizowane jest w warstwie fizycznej. Ze względu na to, że najlepiej zdefiniowane są standardy interfejsów dla tej warstwy, bardzo łatwo można dołączyć w tym miejscu sprzęt szyfrujący, który może być stosowany do każdego typu cyfrowego łącza komunikacyjnego. Ponieważ urządzenia te szyfrują wszystkie przechodzące przez nie dane, w dowolnych inteligentnych węzłach przełączających lub zapamiętujących, między nadawcą a odbiorcą musi nastąpić ich odszyfrowanie. Taki rodzaj szyfrowania jest bardzo korzystny, gdyż kryptoanalitycy nie mogą uzyskać żadnej informacji o strukturze przesyłanych danych. Nie wiadomo, kto z kim rozmawia, jak długie są przesyłane wiadomości, o jakiej porze nawiązywane jest połączenie itd. Ten

poziom zabezpieczenia nazywany jest *ochroną strumienia ruchu komunikacyjnego* (traffic-flow security). Bezpieczeństwo takiego systemu zależy wyłącznie od wybranego algorytmu, a nie od technik zarządzania ruchem. Należy jednak wziąć pod uwagę to, że musi być szyfrowane każde fizyczne łącze w sieci komputerowej, a każdy węzeł w sieci musi być chroniony podczas przetwarzania danych nieszyfrowanych, co w wypadku dużych sieci znacznie zwiększa koszty systemu ochrony danych.

Ze względu na to, że

- szyfrowanie w węzłach końcowych znacznie zmniejsza zagrożenie danych nieszyfrowanych w węzłach sieci,
- szyfrowanie każdego łącz fizycznego uniemożliwia jakąkolwiek analizę informacji dotyczącej kierowania ruchem,

połączenie tych dwóch metod jest najkorzystniejszym sposobem zabezpieczenia sieci, przy czym zarządzanie kluczami może być całkowicie oddzielne.

LITERATURA

[1] Stokłosa J.: Algorytmy kryptograficzne. OWN, Poznań 1994.

[2] Shanon C.E.: Communication theory of secresy systems. Bell System Technical Journal, vol. 28, 1949, 656-715, także w Computer Security Journal, vol. 6, No. 2, 1990, 7-66.

[3] Hellman M.E.: An extension of the Shanon theory approach to cryptography. IEEE Transaction on Information Theory, vol. IT-23, No.#, 1977, 289-294.

[4] Schneier B.: Kryptografia dla praktyków. Protokoły, algorytmy źródłowe w języku C. Wydawnictwa Naukowo-Techniczne, Warszawa 1995.

[5] Schneier B.: Ochrona poczty elektronicznej. Jak chronić prywatność korespondencji w sieci Internet. Wydawnictwo Naukowo-Techniczne, Warszawa 1996.

[6] Rączkiewicz M.: Bezpieczeństwo sieci komputerowych. Wydawnictwo Funkcji Postępu Telekomunikacji, Kraków 1995.

Recenzent: Prof. dr hab. inż. Andrzej Grzywak

Wpłynęło do Redakcji 29 listopada 1996 r.

Abstract

This publication is preliminary work on Cryptographic systems in computer science. It consists of three parts. The first part describes cryptographic systems varying in accessibility of encryption and decryption key, that is:

- private systems
- public systems

as well as the way the algorithm translates open text to cryptogram:

- pipeline type algorithms
- block type algorithms

The second part describes two cryptographic algorithms

- DES
- IDEA

The last part of the publication deals with data protection problems on the level of:

- presentation layer
- transportation layer
- physical layer.