SLOVENSKÁ REPUBLIKA

(19) **SK**

ÚRAD
PRIEMYSELNÉHO
VLASTNÍCTVA
SLOVENSKEJ REPUBLIKY

ZVEREJNENÁ PRIHLÁŠKA VYNÁLEZU

(21) Číslo dokumentu:

1247-2000

(22) Dátum podania prihlášky: **19. 6. 1998**

(31) Číslo prioritnej prihlášky: **98103646, 98104851 98107784**

(32) Dátum podania prioritnej prihlášky: **24. 2. 1998 20. 3. 1998, 22. 4. 1998**

(33) Krajina alebo regionálna organizácia priority: **RU, RU, RU**

(40) Dátum zverejnenia prihlášky: **10. 5. 2001** Vestník ÚPV SR č.: **05/2001**

(62) Číslo pôvodnej prihlášky v prípade vylúčenej prihlášky:

(86) Číslo podania medzinárodnej prihlášky podľa PCT: **PCT/RU98/00181**

(87) Číslo zverejnenia medzinárodnej prihlášky podľa PCT: **WO99/44330**

(13) Druh dokumentu: **A3**

(51) Int. Cl.[7] :

H04L 9/00

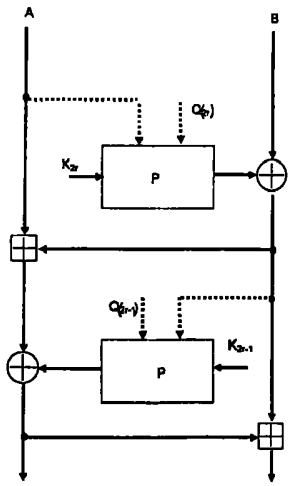
(71) Prihlasovateľ: **Otkrytoe aktsionernoe obschestvo „Moskovskaya Gorodskaya Telefonnaya Set", Moscow, RU;**
Moldovyan Alexandr Andreevich, Vsevolozhsk, RU;
Moldovyan Nikolay Andreevich, Vsevolozhsk, RU;
Savlukov Nikolay Viktorovich, Moskva, RU;

(72) Pôvodca: **Moldovyan Alexandr Andreevich, Vsevolozhsk, RU;**
Moldovyan Nikolay Andreevich, Vsevolozhsk, RU;
Savlukov Nikolay Viktorovich, Moskva, RU;

(74) Zástupca: **PATENTSERVIS BRATISLAVA, a. s., Bratislava, SK;**

(54) Názov: **Spôsob blokového kódovania diskrétnych dát**

(57) Anotácia:
Spôsob patrí do oblasti elektronickej komunikácie a počítačovej technológie, presnejšie sa týka kryptografických spôsobov a zariadení na kódovanie digitálnych dát. Spôsob obsahuje vytvorenie kódovacieho kľúča formou súboru čiastkových kľúčov, rozdelenie dátového bloku na určitý počet čiastkových blokov $N \geq 2$ a striedavé prevádzanie blokov vykonávaním dvojitých operácií na čiastkovom bloku a čiastkovom kľúči. Tento spôsob je charakteristický tým, že pred vykonaním dvojitej operácie na i-tom čiastkovom bloku a čiastkovom kľúči je vykonaná prevodová operácia na čiastkovom kľúči v závislosti od j-teho čiastkového bloku, kde $j \neq i$. Prevodová operácia závislá od j-teho čiastkového bloku je permutačná operácia na bitoch čiastkového kľúča závislá od j-teho čiastkového bloku. Prevodová operácia závislá od j-teho čiastkového bloku je cyklická operácia odsadenia bitov čiastkového kľúča v závislosti od j-tého čiastkového bloku. Prevodová operácia závislá od j-teho čiastkového bloku je substitučná operácia na bitoch čiastkového kľúča v závislosti od j-tého čiastkového bloku.



SK 1247-2000 A3

Spôsob blokového kódovania diskrétnych dát

Oblasť techniky

Predložený vynález sa týka oblasti elektronických komunikácií a počítačových technológií , a presnejšie sa týka spôsobov kódovania na kódovane správ (informácií).

Doterajší stav techniky

V súhrnných vlastnostiach nárokovaného spôsobu sú používané nasledovné pojmy:

- tajný kľúč predstavuje kombináciu bitov známu len legitímnemu užívateľovi;
- kódovací kľúč je bitová kombinácia používaná v kódovacích dátových informačných signáloch; kódovací kľúč je kódovací premenný prvok a používaný je na prevádzanie danej správy alebo danej skupiny správ; kódovací kľúč je vytvorený podľa určených procedúr a tajného kľúča; pri určitom počte šifier je používaný tento tajný kľúč;
- šifra je súhrn elementárnych krokov prevodu vstupných dát použitím kódovacieho kľúča; šifra môže byť realizovaná ako počítačový program alebo ako samostatné elektronické zariadenie;
- čiastkový kľúč je súčasťou kódovacieho kľúča používanou pri jednotlivých elementárnych kódovacích krokoch;
- šifrovanie je proces realizovania určitého spôsobu prevodu dát využívajúceho kódovací kľúč na prevod dát do kryptogramu, ktorý predstavuje pseudonáhodnú sekvenciu znakov, z ktorej je prakticky nemožné získať informácie bez znalosti hodnoty kódovacieho kľúča;
- dešifrovanie je reverzný proces vzhľadom na šifrovací proces; dešifrovanie zabezpečuje obnovenie informácie podľa kryptogramu, ak je známy kódovací kľúč;

- kryptografická odolnosť je miera bezpečnosti ochrany dát a reprezentuje intenzitu práce meranú podľa počtu elementárnych operácií, ktoré musia byť vykonané s cieľom obnovenia informácie z kryptogramu ak je známy prevodný algoritmus, avšak bez znalosti kódovacieho kľúča.

Známe sú spôsoby blokového dátového kódovania, viď. napr. šifra RC5 [R.Rivest, The RC5 Encryption Algorithm, Fast Software Encryption, second International Workshop Proceendings (Leuven, Belgie, December 14-16, 1994), Lecture Notes in Computer Science, v.1008, Springer-Verlag, 1995, str. 86-96]. Podľa známeho spôsobu je dátové blokové kódovanie uskutočnené vygenerovaním kódovacieho kľúča v tvare skupiny čiastových kľúčov, rozdelením prevádzaného dátového bloku na čiastkové bloky a striedavou zmenou uvedeného použitím cyklickej odsadzovacej operácie, modulo 2 súčtovej operácie vykonanej na dvoch čiastkových blokoch, a modulo 2^{32} súčtovej operácie vykonanej na čiastkovom bloku a čiastkovom kľúči. V tomto prípade sú čiastkové kľúče použité podľa pevného programu, t.j. v danom kroku uskutočnenia binárnej operácie medzi čiastkovým blokom a čiastkovým kľúčom hodnota čiastkového kľúča nezávisí na dátovom vstupnom bloku. Tento spôsob blokového kódovania zabezpečuje vysokú kódovaciu rýchlosť, ak je realizovaný ako počítačový program.

Tento spôsob však zlyháva pri zabezpečení dostatočnej odolnosti voči diferenciálnej a lineárnej kryptoanalýze [Kalisky B.S., Yin Y.L. On Differential and Linear Cryptanalysis of the RC5 Encryption Algorithm. Advanced in Cryptology – CRYPTO´ 95 Proc., Springer-Verlag, 1995, str. 171 - 184], čo je spôsobené tým, že pri tomto spôsobe sú používané v daných kódovacích krokoch pevné čiastkové kľúče pre všetky možné vstupné bloky.

Nárokovanému blokovému kódovaciemu spôsobu je vo svojej technickej podstate najbližším spôsob opísaný v US štandarte DES [National Bureau of Standarts. Data Encryption Standart. Federal Information Proceendings Standard Publication 46, January 1977]. Tento spôsob obsahuje vygenerovanie kódovacieho kľúča v tvare súboru 48 – bitových čiastkových kľúčov, rozdelenie vstupného bloku diskrétnych dát na dva 32 – bitové čiastkové bloky L a R,

a striedavá premena čiastkových blokov na riadenie tajným kľúčom. Celkove je vykonaných 16 cyklov prevodu 32-bitového cyklus prevodu čiastkového bloku je uskutočnený vykonaním čiastkového bloku. Každý cyklus prevodu čiastkového bloku je uskutočnený vykonaním nasledovných procedúr: (1) rozšírenie čiastkového bloku R na 48 bitov tohto čiastkového bloku: R->R´, (2) vyhotovenie modulo 2 súčtovej operácie na čiastkovom bloku a čiastkovom kľúči, rozdelenie čiastkového bloku R´ na osem 6-bitových čiastkových blokov, (4) vykonanie substitučnej operácie na každom 6-bitovom čiastkovom bloku nahradením 6-bitových čiastkových blokov 4-bitovými čiastkovými blokmi podľa známych substitučných tabuliek, (5) kombinovanie ôsmich 4-bitových čiastkových blokov do 32-bitového čiastkového bloku 2, (6) vykonanie operácie bitovej permutácie R čiastkových blokov podľa určeného pravidla, (7) vykonanie modulo 2 súčtovej operácie na čiastkovom bloku R s čiastkovým blokom L. Pri vykonávaní aktuálneho kódovacieho cyklu je používaný pevný čiastkový kľúč na všetky možné dátové vstupné bloky. Čiastkové kľúče používané pri prevode čiastočných blokov sú vygenerované riadením 56-bitového tajného kľúča. Tento spôsob informačného blokového kódovania dosahuje veľké prevodové rýchlosti, ak je realizovaný pomocou špeciálneho elektronického obvodu.

Tento spôsob má však niektoré nevýhody, menovite dosahuje nízke kódovacie rýchlosti, ak je realizovaný pomocou software. Ďalej, tento spôsob využíva 56-bitový tajný kľúč, čo umožňuje výkonným moderným počítačom odhaliť tajný kľúč výberom možných kľúčových hodnôt. Toto vyžaduje vykonanie niekoľko kódovacích procedúr využívajúcich rôzne tajné kľúče, čo robí spôsob obtiažnym na získanie vysokej kódovacej rýchlosti aj v prípade hardwarovej realizácie.

Základom vynálezu je úloha vyvinúť spôsob blokového kódovania diskrétnych dát, kedy prevod dátového čiastkového bloku bude ovplyvňovaný tak, aby bol znížený počet prevodových operácií uvažovaných na jeden vstupný dátový bit, pričom bude zároveň poskytovať vysokú kryptografickú odolnosť, čo bude viesť ku zvýšenej kódovacej rýchlosti.

Podstata vynálezu

Vyššie spomenutý cieľ je dosiahnutý faktom, že spôsob blokového kódovania diskrétnych dát zahrňuje vygenerovanie kódovacieho kľúča ako súbor čiastkových kľúčov, rozdelenie dátového bloku na N>=2 čiastkových blokov a striedavú premenu čiastkového bloku vykonaním dvojitej operácie na čiastkovom bloku a čiastkovom kľúči, pričom novou vlastnosťou podľa vynálezu je vykonanie j-tej blokovo-závislej čiastkovej operácie, kde j≠1, na čiastkovom kľúči pred vykonaním dvojitej operácie na i-tom čiastkovom bloku a čiastkovom kľúči. Vďaka tomuto riešeniu závisí štruktúra čiastkového kľúča používaná v danom kódovacom kroku na vykonávaných dátach a tak sú v danom prevodnom kroku použité odlišné modifikované hodnoty čiastkového kľúča pre odlišné vstupné bloky, vďaka čomu je poskytovaná vysoká kryptografická odolnosť voči diferenciálnej kryptoanalýze, pričom je zároveň znížený počet kódovacích cyklov a to vedie ku zvýšeniu rýchlosti kryptografického prevodu.

Novou vlastnosťou je tiež fakt, že ako j-tá blokovo-závislá prevodná operácia je využitá bitová permutačná operácia j-tého blokovo-závislého čiastkového kľúča.

Vďaka tomuto riešeniu je poskytovaná zvýšená kódovacia rýchlosť, ak je nárokovaný spôsob realizovaný formou elektronického kódovacieho zariadenia.

Novou vlastnosťou je tiež to, že bitová cyklická odsadzovacia operácia j-tého blokovo-závislého čiastkového kľúča je použitá ako j-tá blokovo-závislá prevodná operácia.

Vďaka tomuto riešeniu je poskytovaná zvýšená kódovacia rýchlosť, ak je nárokovaný spôsob realizovaný ako počítačový kódovací software.

Ďalšou novou vlastnosťou je to, že j-tá blokovo závislá permutačná operácia s čiastkovým kľúčom je využitá ako j-tá blokovo závislá prevodná operácia.

Vďaka tomuto riešeniu je poskytnuté dostatočné zvýšenie kódovacej kryptografickej odolnosti, zároveň zabezpečujúcej vysokú kódovaciu rýchlosť, ak je nárokovaný spôsob realizovaný ako počítačový kódovací software.

Nižšie bude podstata nárokovaného vynálezu opísaná detailnejšie na vyhotoveniach s odkazmi na pripojené obrázky.

Prehľad obrázkov na výkresoch

Obr.1 predstavuje zovšeobecnený kódovací diagram podľa nárokovaného spôsobu.

Obr.2 predstavuje blokový diagram elementárne riadeného spínača, ktorým je základný prvok riadeného permutačného bloku. Ak je u = 1, vstupné bity nie sú permutované, t.zn. výstupné signály sa zhodujú so vstupnými signálmi. Ak je u = 0, potom sú vstupné bity permutované.

Obr.3 predstavuje tabuľku vstupných a výstupných signálov elementárne riadeného spínača, ak má potenciál riadiaceho signálu vysokú hodnotu (high).

Obr.4 predstavuje tabuľku vstupných a výstupných signálov elementárne riadeného spínača, ak má potenciál riadiaceho signálu nízku hodnotu (low).

Obr.5 schématicky znázorňuje štruktúru riadeného permutačného bloku, pozostávajúceho zo súboru blokov rovnakého typu, elementárnych spínačov, ktorý realizuje 2^{79} rôznych permutácií vstupných bitov v závislosti od hodnoty 79-bitového riadiaceho kódu.

Obr.6 predstavuje diagram zjednodušeného riadeného permutačného bloku.

Príklady uskutočnenia vynálezu

Vynález je vysvetlený pomocou zovšeobecneného diagramu kryptografického prevodu dátového bloku založeného na nárokovanom spôsobe, ktorý je predstavený na obr.1, kde P je blok riadenej operácie vykonanej na čiastkovom kľúči; A a B sú prevedené n-bitové čiastkové bloky; K_{2r}, K_{2r-1} sú m-bitové čiastkové kľúče (všeobecne $m \neq n$); Q(2r), Q(2r-1) sú g-bitové dodatočné čiastkové kľúče; znamienko „⊕" označuje modulo 2 bit-za-bitom súčtovú operáciu; znamienko „⊗" označuje modulo 2^n súčtovú operáciu. Hrubé neprerušované čiary označujú n-bitovú signálovú prenosovú zbernicu, tenké

bodkované čiary označujú prenos jedného riadiaceho bitu. Hrubé bodkované čiary označujú zbernicu na prenos n riadiacich signálov, s ktorými sú použité prevádzané bity čiastkového bloku. Hrubé bodkované čiary označujú tiež zbernicu na prenos h bitov dodatočných čiastkových kľúčov $\underline{Q(2r)}$ a $\underline{Q(2r-1)}$, ktoré slúžia na zmenu operácie v závislosti na čiastkovom bloku, ktorý je prevádzaný. V konkrétnych prípadoch nemusia byť použité dodatočné čiastkové kľúče.

Obr.1 znázorňuje jednotlivý (r-tý) kódovací cyklus. V závislosti od presného typu použitej riadiacej operácie a od požadovanej prevodovej rýchlosti môže byť nastavených od 6 do 10 alebo viacej cyklov. Jednotlivý prevodový cyklus obsahuje vykonanie nasledovnej sekvencie procedúr:

(1) prevod čiastkového kľúča $\underline{K_{2r}}$ v závislosti od hodnoty čiastkového bloku $\underline{A}$ a od hodnoty dodatočného čiastkového kľúča $\underline{Q(2r)}$, ktorého výsledkom je vygenerovanie prevedenej hodnoty čiastkového kľúča $\underline{P_{A,Q(2r)}(K_{2r})}$ na výstupe bloku $\underline{P_1}$;

(2) prevod čiastkového bloku $\underline{B}$ vykonaním modulo 2 bit-za-bitom súčtovej operácie na hodnote $\underline{P_{A,Q(2r)}(K_{2r})}$ a čiastkového bloku $\underline{B}$: B:= B⊕ $P_{A,Q(2r)}(K_{2r})$, kde znamienko „:=" označuje operáciu priradenia;

(3) prevod čiastkového bloku $\underline{A}$ vykonaním modulo 2^n súčtovej operácie na čiastkovom bloku $\underline{A}$ a čiastkovom bloku $\underline{B}$: A:=A⊗B;

(4) prevod čiastkového kľúča $\underline{K_{2R-1}}$ v závislosti od hodnoty čiastkového bloku $\underline{B}$ a od hodnoty dodatočného čiastkového kľúča $\underline{Q(2r-1)}$ ktorého výsledkom je vygenerovanie prevedenej hodnoty čiastkového kľúča $\underline{P_{A,Q(2r-1)}(K_{2r-1})}$ na výstupe bloku $\underline{P_2}$;

(5) prevod čiastkového bloku $\underline{A}$: A:= A⊕ $P_{A,Q(2r-1)}(K_{2r-1})$;

(6) prevod čiastkového bloku $\underline{B}$: B:=B⊗A.

V závislosti od konkrétneho uskutočnenia navrhovaného spôsobu blokového kódovania diskrétnych informácií môže byť pri uskutočnení jednotlivých kódovacích cyklov použitý rovnaký pár m-bitových čiastkových kľúčov $\underline{K_2}$ a $\underline{K_1}$ (dodatočné g-bitové čiastkové kľúče $\underline{Q(2)}$ a $\underline{Q(1)}$. Možné je uskutočnenie, kedy sú pri jednotlivých cykloch použité nezávislé čiastkové kľúče $\underline{K_{2r}}$ a $\underline{K_{2r-1}}$ ($\underline{Q(2r)}$

a Q(2r-1)). Ak je napríklad počet cyklov r=3, prvý cyklus používa čiastkové kľúče K_2 a K_1 (Q(2) a Q(1)), druhý cyklus používa čiastkové kľúče K_2 a K_3 (Q(4) a Q(3)), tretí cyklus používa čiastkové kľúče K_6 a K_5 (Q(6) a Q(5)). Čiastkové kľúče K_{2r}, K_{2r-1} a dodatočné čiastkové kľúče Q(2r) a Q(2r-1) môžu byť vytvorené podľa špeciálnych procedúr v závislosti od tajného kľúča. Možné je uskutočnenie, pri ktorom sú čiastkové kľúče K_{2r} a K_{2r-1} a dodatočné čiastkové kľúče Q(2r) a Q(2r-1) vytvorené vygenerovaním náhodných pravidiel.

Možné technické uskutočnenie nárokovaného spôsobu je vysvetlené pomocou jeho nasledujúcich špecifických uskutočnení.

Príklad 1.

Tento príklad vysvetľuje kódovanie 64-bitových dátových blokov využívajúci riadené permutácie ako operácie vykonané na čiastkovom kľúči v závislosti od jedného z blokov, ktorý je prevádzaný. Kódovací kľúč je vygenerovaný ako 16 čiastkových kľúčov K_1, K_2, K_3, ... K_{16}, z ktorých každý má dĺžku 32 bitov. Dodatočné čiastkové kľúče nie sú zavedené. Dátový vstupný blok je rozdelený na dva 32-bitové čiastkové bloky A a B. Kódovanie vstupného bloku je opísané nasledovným algoritmom:

1. Nastavenie čísla cyklu:

r:=1.

2. Prevod čiastkového bloku B podľa vzťahu:

$B:= B \oplus P_A(K_{2r})$,

kde $P_A(K_{2r})$ označuje operáciu permutácie bitov čiastkového kľúča K_{2r} vykonanou v závislosti od hodnoty čiastkového bloku A.

3. Prevod čiastkového bloku A podľa vzťahu:

$A:=A \otimes B$.

4. Prevod čiastkového bloku A podľa vzťahu:

$A:= A \oplus P_B(K_{2r-1})$,

kde $P_B(K_{2r-1})$ označuje operáciu permutácie bitov čiastkového kľúča K_{2r-1} vykonanou v závislosti od hodnoty čiastkového bloku B.

5. Prevod čiastkového bloku B podľa vzťahu:

B:=B⊗A.

6. Ak r≠8, pripočítaj jednotku k číslu r:=r+1

a prejdi na krok 2, inak STOP.

Tento algoritmus je orientovaný na realizáciu formou elektronického obvodu. Operácia bitových permutácií čiastkových kľúčov v závislosti od jedného z prevádzaných čiastkových blokov môže byť vykonaná použitím riadeného permutačného bloku realizovaného na základe použitia súboru elementárnych spínačov, ktoré vykonávajú operáciu permutácie dvoch bitov.

Obr. 2 vysvetľuje operáciu elementárneho spínača, kde u je riadiaci signál, a a b sú dátové vstupné signály, c a d sú dátové výstupné signály.

Tabuľky na obr.3 a obr.4 znázorňujú závislosť výstupného signálu od vstupného signálu a riadiacich signálov. Z týchto tabuliek bude zrejmé, že pri u=1 sa riadok a mení s riadkom c a riadok b s riadkom d. Ak je u=0, potom sa riadok a mení s riadkom d a riadok b s riadkom c. Ak je potom riadiaci signál rovnajúci sa jednej, žiadne dva vstupné bity nie sú permutované, ak je však riadiaci signál rovný nule, vstupné bity sú permutované.

Obr.5 znázorňuje možné uskutočnenie riadeného permutačného bloku využívajúceho súbor elementárnych spínačov S. Tento príklad zodpovedá bloku P obsahujúceho 32-bitový informačný vstup a 79-bitový riadiaci vstup. Bity aktuálne prevádzaného čiastkového kľúča sú použité ako informačné signály. 32 bitov jedného z čiastkových blokov a 47 bitov jedného z dodatočných čiastkových kľúčov je použitých ako riadiace signály.

Počet možných verzií permutačných operácií je rovný počtu možných kódových kombinácií na riadiacom vstupe a pre blok P so štruktúrou zobrazenou na obr.2 je 2^{79}. Tento riadený permutačný blok realizuje jedinečnú permutáciu vstupných bitov pre každú možnú hodnotu kódovej kombinácie na riadiacom vstupe, ktorých počet je 2^{79}. Ďalšie informačné vstupy riadeného permutačného bloku sú označené i1, i2, ...,i32, vonkajšie výstupy sú označené o1, o2, ... , o32, riadiace výstupy sú označené c1, c2, ... , c79. Elementárne spínače S sú spojené tak, aby vytvorili pole pozostávajúce z 31 riadkov. V prvom riadku je zapojených 31 elementárnych spínačov S, v druhom riadku 30

spínačov, v treťom riadku 29 spínačov, atď. V každom nasledujúcom riadku je počet elementárnych spínačov znížený o 1. V najnižšom 31. riadku je zapojený jeden elementárny spínač.

Riadok označený j≠31 má 33-j výstupov a 32-j riadiacich vstupov. Posledný (najviac napravo) vstup j-tého riadku je vonkajší výstup riadeného permutačného bloku, zostávajúcich 32-j výstupov j-tého riadku je pripojených na odpovedajúce vstupy (j+1)-tého riadku. Posledný 31. Riadok má dva výstupy a obidva z nich sú vonkajšie výstupy riadeného permutačného bloku. Len na jeden riadiaci vstup každého riadku je aplikovaný jednotkový (u=1) riadiaci signál. Na splnenie týchto požiadaviek sú poskytnuté dva-tridsať-dva-stupňové dekódovače $\underline{F}_1$, $\underline{F}_2$, ... , $\underline{F}_{15}$, a dva-šesťnásť-stupňový dekódovač $\underline{F}_{16}$. Dekódovače $\underline{F}_1$, $\underline{F}_2$, ... , $\underline{F}_{15}$ majú päť vonkajších riadiacich vstupov, ku ktorým je privedený náhodný 5-bitový binárny kód, a 32 výstupov. Tieto dekódovače generujú len na jednom výstupe jednotkový signál. Na zostávajúcich 31 výstupoch je nastavený nulový signál. Dekódovač $\underline{F}_{16}$ má 4 vstupy, ku ktorým je privedený ľubovoľný 4-bitový binárny kód, a 16 výstupov, z ktorých len na jednom je nastavená hodnota signálu jedna. Pre všetky dekódovače $\underline{F}_1$, $\underline{F}_2$, ... , $\underline{F}_{15}$ a $\underline{F}_{16}$ každá vstupná binárna kódová hodnota nastavuje jediné možné výstupné číslo, pri ktorom je nastavený jednotkový signál (u=1).

Časť výstupov kódovača $\underline{F}_h$, kde h≤15, je pripojená na riadiace vstupy riadku očíslovaného h (32-h výstupov), pokým časť výstupov je pripojená na riadiace vstupy (32-h)-tého riadku (h-výstupov). Preto je v každom riadku nastavený riadiaci signál u=1 len na jednom elementárnom spínači. Riadkový vstup spojený s pravým vstupom elementárneho spínača, na ktorý je aplikovaný jednotkový riadiaci signál, komutuje s vonkajším výstupom riadeného permutačného bloku odpovedajúceho danému riadku. Ak je jednotkový riadiaci signál aplikovaný na elementárny spínač umiestnený najviac naľavo, vonkajší výstup riadeného permutačného bloku (blok) komutuje s riadkovým vstupom umiestneným najviac naľavo. Prvý riadok komutuje jeden z vonkajších vstupov $\underline{i1}$, $\underline{i2}$, ... , $\underline{i32}$ bloku $\underline{P}$ s vonkajším výstupom $\underline{o1}$ a zvyšných 31 vonkajších vstupov so vstupmi druhého riadku. Druhý riadok komutuje jeden zo zvyšných 31

vonkajších vstupov s vonkajším výstupom o2 a zvyšných 30 vonkajších vstupov so vstupmi tretieho riadku, atď. Táto štruktúra bloku P implementuje jedinú permutáciu vstupných bitov pre všetky hodnoty binárneho kódu privedených na 79-bitový riadiaci vstup bloku P.

Vykonateľná je nasledovná verzia použitia riadiaceho permutačného bloku P s 32-bitovým informačným vstupom a 79-bitovým riadiacim vstupom. 32 bitov čiastkového bloku A a 47 bitov dodatočného 47-bitového čiastkového kľúča Q(2r) môžu sa použiť ako riadiace signály aplikované na 79-bitový riadiaci vstup riadeného permutačného bloku P. V tomto prípade, v závislosti na 47-bitovom dodatočnom čiastkovom kľúči, je vytvorená jedna z 2^{47} rôznych modifikácií bitovej permutačnej operácie, ktorá závisí od hodnoty vstupného bloku. Pritom každá modifikácia tejto operácie obsahuje 2^{32} rôznych operácií permutácie bitov čiastkového kľúča K_{2r}, kde výber určitej permutačnej operácie je určený hodnotou čiastkového bloku A. Výber modifikácie nie je preddefinovaný, pretože je definovaný dodatočným čiastkovým kľúčom Q(2r), ktorý je priamo prvkom tajného kľúča alebo závisí od tajného kľúča. Toto ďalej zvyšuje odolnosť voči kryptografickému prevodu. Ak kódovacie zariadenie používa dva bloky P ktoré majú štruktúru zobrazenú na obr.2, počet možných modifikovaných kombinácií riadenej permutačnej operácie nastavenej na blokoch P v závislosti na dodatočných 47-bitových čiastočných kľúčov môže byť nastavený až do $(2^{47})^2=2^{94}$, ak je použitý tajný kľúč s dĺžkou 94 bitov.

Moderná technológia výroby integrovaných obvodov umožňuje vďaka jednoduchej štruktúre blokov P jednoduchú výrobu kryptografických mikroprocesorov obsahujúcich riadené permutačné bloky so vstupnou kapacitou 32 a 64 bitov a poskytujúcich kódovaciu rýchlosť až do 1 Gbit/s a vyššiu.

Obr.6, na ktorom tenké plné čiary označujú prenos jedného bitu čiastkového kľúča, poukazuje na možnú realizáciu riadeného permutačného bloku využívajúceho súbor elementárnych spínačov S. Tento prípad riadeného permutačného bloku sa zhoduje s riadeným permutačným blokom. Obsahuje 8-bitový vstup pre informačné signály (bity čiastkového kľúča) a 8-bitový vstup pre riadiace signály (bity dátového čiastkového bloku označené bodkovanými čiarami

podobným čiaram na obr.1). Podobným spôsobom je možné vytvoriť ľubovoľný riadený permutačný blok, obsahujúci napríklad 64-bitový vstup pre informačné signály a 128-bitový vstup pre riadiace signály. Používaním riadeného permutačného bloku s 32-bitovým informačným vstupom je počet rôznych permutácií rovný 2^{32}. To znamená, že pri kódovaní dvoch rôznych dátových blokov je možnosť opakovania určitej permutácie v danom súbore rovná 2^{-32}, zatiaľ čo možnosť opakovania permutácií v z súborových krokoch je rovná 2^{-32z}. Súbor pozmenených hodnôt čiastkového kľúča použitého na prevod každej vstupnej správy je preto prakticky jediná, čo zabezpečuje vysokú kryptografickú odolnosť kódovania.

Použitím zjednodušenej štruktúry riadeného permutačného bloku zobrazeného na obr.6 je jednoduché vyrobiť kryptografické mikroprocesory obsahujúce riadené permutačné bloky so vstupnou kapacitou až do 128 bitov. Použitie riadených permutačných operácií na 128-bitovom čiastkovom kľúči umožňuje získať vyššiu kryptografickú odolnosť kódovania. Riadený permutačný blok je kombinačný elektrický obvod, ktorý poskytuje vysokú rýchlosť vykonania riadených permutácií.

Príklad 2.

Tento príklad vysvetľuje použitie cyklickej oddeľovacej operácie závislej od čiastkových blokov, ktoré sú vykonávané na čiastkových kľúčoch. Kódovací kľúč je generovaný v tvare 16 čiastkových kľúčov $\underline{K}_1$, $\underline{K}_2$, $\underline{K}_3$, ... $\underline{K}_{32}$, z ktorých každý má dĺžku 32 bitov. Vstupný 64-bitový dátový blok je rozdelený na dva 32-bitové čiastkové bloky $\underline{A}$ a $\underline{B}$. Kódovanie vstupného bloku je znázornené nasledovným algoritmom:

1. Nastavenie čísla cyklu:

r:=1.

2. Prevod čiastkového bloku $\underline{B}$ podľa vzťahu:

$B := B \oplus (K_{2r} <<< A)$,

kde $K_{2r} <<< A$ označuje operáciu cyklického oddelenia naľavo pomocou $\underline{A}$ bitov vykonanou na čiastkovom kľúči K_{2r}.

3. Prevod čiastkového bloku A podľa vzťahu:

$A:=A \otimes B$.

4. Prevod čiastkového bloku A podľa vaťahu:

$A:= A \oplus (K_{2r-1} <<< B)$,

kde $K_{2r-1} <<< B$ označuje operáciu cyklického oddelenia naľavo pomocou B bitov vykonanou na čiastkovom kľúči K_{2r-1}.

5. Prevod čiastkového bloku B podľa vzťahu:

$B:=B \otimes A$.

6. Ak $r \neq 16$, pripočítaj jednotku k číslu $r:=r+1$

a prejdi na krok 2, inak STOP.

Logický vzor prevodového cyklu je vysvetlený na obr.1, kde bloky P_1 a P_2 v tomto príklade predstavujú operačný blok vykonávajúci operáciu cyklického bitov odpovedajúceho čiastkového kľúča v závislosti od čiastkových blokov, ktoré sú prevádzané. Tento algoritmus je orientovaný na realizáciu formou počítačového programu. Moderný mikroprocesor rýchle vykonáva cyklické oddeľovacie operácie v závislosti na hodnote premennej uloženej v jednom z registrov. Vďaka tomuto faktu poskytuje opísaný algoritmus, ak je realizovaný softwarovo, kódovacia rýchlosť okolo 40 Mbit/s pre rozšírený mikroprocesor Pentium/200. Ak je nastavených 10 kódovacích cyklov, dosiahne sa rýchlosť okolo 60 Mbit/s.

Príklad 3.

Tento príklad vysvetľuje použitie substitučných operácií závislých na čiastkových blokoch, ktoré sú oddeľované, a vykonávaných na čiastkových kľúčoch. V predmetnom príklade predstavujú bloky P_1 a P_2 operačný blok vykonávajúci substitučnú operáciu v závislosti na zodpovedajúcich čiastkových blokoch. Pod pojmom substitučná operácia sa rozumie operácia nahradenia binárnej hodnoty signálu na vstupe operačného bloku P inou binárnou hodnotou (nastavenou na výstupe operačného bloku), ktorá je vybraná v závislosti od hodnoty na vstupe bloku P v súlade s určitou vstupnou tabuľkou. Môžu sa realizovať dve substitučné verzie:

(1) n-bitový vstupný binárny vektor je nahradený n-bitovým výstupným binárnym vektorom, kde rôznym vstupným binárnym vektorom odpovedajú rôzne výstupné binárne vektory;

(2) m-bitový binárny vektor je nahradený n-bitovým binárnym vektorom, kde $n \neq m$, pričom rôznym vstupným binárnym vektorom môžu odpovedať jednak rozdielne tak aj zhodné výstupné binárne vektory.

Vysvetlime si teraz špecifikujúcu závislosť prvého typu substitučnej operácie na čiastkovom bloku dát, ktorý sa vykonáva. Predpokladajme, že substitučné operácie sú vykonávané na binárnych vektoroch s n-bitovou dĺžkou, kde n je celé číslo, na zabezpečenie substitučnej operácie s kapacitou nxn (označenie nxn označuje, že vstupom pre substitučnú operáciu je binárny vektor s dĺžkou n bitov, a výstupný binárny vektor má tak isto dĺžku n bitov). Potrebné je použiť tabuľku obsahujúcu dva riadky čísiel:

0	1	2	3	. . .	N-1
α_0	α_1	α_2	α_3	. . .	α_{N-1}

kde $N=2^n$. V spodnom riadku tejto tabuľky sa nachádzajú všetky možné hodnoty n-bitového bloku zodpovedajúceho raz, avšak v ľubovoľnom poradí. Správne poradie umiestnenia čísiel v spodnom riadku určuje špecifickú verziu substitučnej tabuľky a preto aj špecifickú verziu substitučnej operácie vykonávanej použitím tejto tabuľky. Vykonávanie substitučnej operácie je urobené nasledovne. V hornom riadku je zvolené číslo rovné hodnote vstupného bloku. Hodnota ktorá sa nachádza pod týmto číslom v spodnom riadku je použitá ako vstupný blok. Substitučná tabuľka môže byť preto umiestnená v pracovnej pamäti počítača ako po sebe nasledujúca sústava n-bitových počítačových slov umiestnených vo vnútri buniek s adresami $\underline{w}_0$, $\underline{w}_1$, $\underline{w}_2$, ... $\underline{w}_{N-1}$. V tomto prípade slúži hodnota vstupného binárneho vektora $\underline{Y}$ na výpočet adresy $\underline{w}_0$ + Aby slova, ktorá je vybraná ako výstup binárneho vektora. Tento spôsob reprezentácie substitučnej tabuľky vyžaduje použitie kapacity rovnajúcej sa $Nn = 2^n n$ bitov. Uvažujme počet substitučných tabuliek rovný 2^L (požadovaná kapacita pamäte bude v tomto prípade $2^L Nn$ bitov) a umiestnenie substitučnej tabuľky neprerušovane jedna za druhou. Zoberieme hodnotu adresy $\underline{w}_0$ z prvého bitu slova tabuľky ako tabuľkovú

adresu s číslom v, Nech tabuľková adresa s číslom v = 0 je s. V tomto prípade substitučná tabuľková adresa s ľubovoľným číslom v je s + vN. Ak je riadiaci binárny vektor špecifikovaný tak, že určuje číslo aktuálnej substitučnej tabuľky rovnako ako aktuálny vstupný binárny vektor, potom je substitučná operácia vykonaná nahradením aktuálneho vstupného bloku n-bitovým slovom umiestnením na adrese s + vN + Y, kde Y je hodnota vstupného binárneho vektora, na ktorom je vykonávaná aktuálna substitučná operácia. Použitím tohto vzťahu je jednoduché špecifikovať výber substitučnej tabuľky s číslom v a vykonať substitúciu na vstupnom binárnom vektore s hodnotou Y. Vzhľadom na to je špecifikovanie závislosti substitučných tabuliek od hodnoty riadiaceho binárneho vektora a vykonávanie substitučnej operácie vykonané mikroprocesorom veľmi rýchlo, ak sú vybrané zodpovedajúce hodnoty parametrov L a n, napríklad ak je L = 5 a n = 8. Ak sú tieto parametre vybrané, potom je pre lokalizáciu substitučnej tabuľky požadovaných 8k bitov pracovnej pamäte, čo je prijateľné, pretože moderné počítače majú kapacitu pracovnej pamäte o mnoho rádov vyššiu než je táto hodnota (od 1 do 64 Mbytov a viac).

Teraz vysvetlíme špecifikujúcu závislosť druhého typu substitučnej operácie na dátovom čiastkovom bloku pomocou príkladu 16x32 substitúcií špecifikovaných použitím číslovanej sekvencie 32-bitových binárnych vektorov X_j, $J = 0,1,2,...,12^{16} - 1$. Predpokladá sa, že sekvencia X_j je známa a týka sa popisu kódovacieho algoritmu. Substitučná operácia na 16-bitovom kľúči k je vykonaná v závislosti na prevádzanom čiastkovom bloku b nasledovne:

(1) je vyrátané číslo $j = (b + k) \bmod 2^{16}$;

(2) 16-bitový binárny vektor k je nahradený 32-bitovým binárnym vektorom x_j;

Kódovanie 64-bitových dátových blokov podľa substitučných operácií využívajúcich sekvenciu 32-bitových binárnych vektorov x_j ($J = 0,1,2,...,12^{16} - 1$) na čiastkových kľúčoch závislých od dátových čiastkových blokov, ktoré sú prevádzané, môže sa uskutočniť napríklad nasledovne. Kódovací kľúč je generovaný formou 16 čiastkových kľúčov K_1, K_2, K_3, ... K_{32}, z ktorých každý má dĺžku 16 bitov. Vstupný dátový blok je rozdelený na dva 32-bitové čiastkové

bloky A = a_2 | a_1 a B = b_2 | b_1 reprezentovaných ako kaskáda príslušných 16-bitových čiastkových blokov a_1 , a_2 a b_1 , b_2. Kódovanie vstupného bloku je podľa nasledovného algoritmu:

1. Nastavenie čísla cyklu r = 1.
2. Prevod čiastkového bloku B podľa vzťahu:

   B:= B⊕ F(K_{4r} , a_1) ,

   kde F(K_{4r} , a_1) označuje substitučnú operáciu na čiastkovom kľúči K_{4r} v závislosti na čiastkovom bloku a_1.
3. Prevod čiastkového bloku A podľa vzťahu:

   A := A + B(mod2^{32}).
4. Prevod čiastkového bloku A podľa vzťahu:

   A := A⊕ F(K_{4r-1} , b_1) ,

   kde F(K_{4r-1} , b_1) označuje substitučnú operáciu na čiastkovom kľúči K_{4r-1} v závislosti na čiastkovom bloku b_1.
5. Prevod čiastkového bloku B podľa vzťahu:

   B := B + A(mod2^{32}).
6. Prevod čiastkového bloku B podľa vzťahu:

   B := B ⊕ F(K_{4r-2} , b_2) ,
7. Prevod čiastkového bloku A podľa vzťahu:

   A := A + B(mod2^{32}).
8. Prevod čiastkového bloku A podľa vzťahu:

   A := A ⊕ F(K_{4r-3} , b_2) ,
9. Prevod čiastkového bloku B podľa vzťahu:

   B := B + A(mod2^{32}).
10. Ak r ≠ 4, pripočítaj jednotku k číslu r := r+1 a prejdi na krok 2, inak STOP.

Tento algoritmus používa známu substitučnú tabuľku veľkosti 240 kbytov, ktorá zaberá malú časť kapacity pracovnej pamäte moderného počítača. Operácia výberu binárnych vektorov z pracovnej pamäte podľa preddefinovaných adries je vykonaná malým počtom strojových cyklov, vďaka čomu je softwarové vyhotovenie navrhovaného spôsobu blokového kódovania so

substitučnými operáciami vykonávanými na čiastkových kľúčoch v závislosti od prevádzaných čiastkových blokov poskytuje kódovaciu rýchlosť od 20 do 60 Mbit/s (v závislosti od špecifického vyhotovenia) pre rozšírený mikroprocesor Pentium/200.

Priemyselná využiteľnosť

Uvedené príklady ukazujú, že navrhovaný spôsob blokového kódovania diskrétnych dát je technicky uskutočniteľný a umožňuje riešenie problému, ktorý sme definovali.

Diskutované príklady sú jednoducho uskutočniteľné, napríklad formou špecializovaných mikroelektronických kódovacích obvodov (Príklad 1) a formou kódovacieho počítačového software (Príklady 2 a 3) a zabezpečujú kódovaciu rýchlosť až do 1 Gbit/s a vyššiu (Príklad 1), ak ide o hardwarovú realizáciu, a až do 60 Mbit/s, ak ide o softwarovú realizáciu a použitím rozšíreného mikroprocesoru Pentium/200 (Príklady 2 a 3).

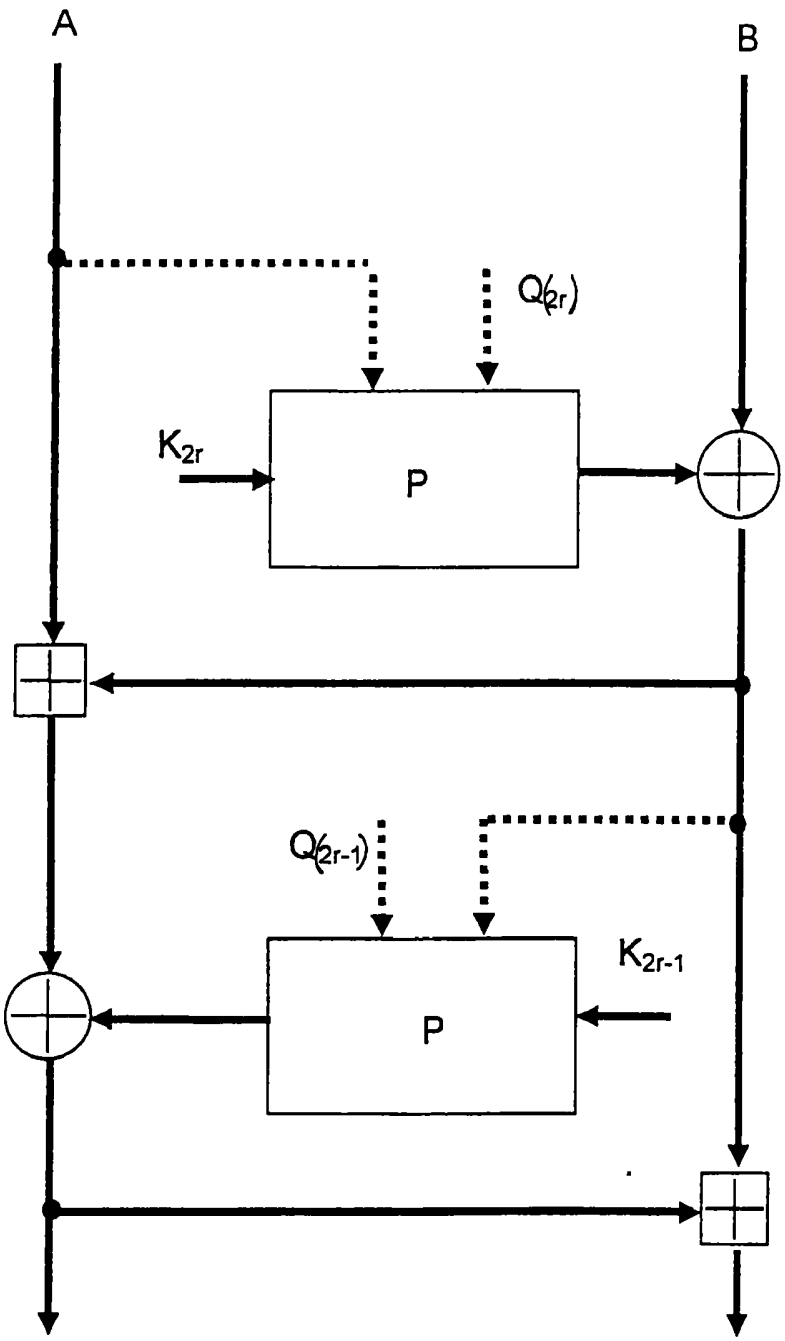
PATENTOVÉ NÁROKY

1. Spôsob blokového kódovania diskrétnych dát, obsahujúci vygenerovanie kódovacieho kľúča formou súboru čiastkových kľúčov, rozdelenie dátového bloku na N ≥ 2 čiastkových blokov a striedavé prevádzanie spomínaných blokov vykonávaním dvojitej operácie na čiastkovom bloku a čiastkovom kľúči, v y z n a č u j ú c i s a t ý m, že pred vykonávaním spomenutej dvojitej operácie na i-tom čiastkovom bloku a čiastkovom kľúči je vykonaná prevodová operácia na čiastkovom kľúči v závislosti od j-teho čiastkového bloku, kde j ≠ i.

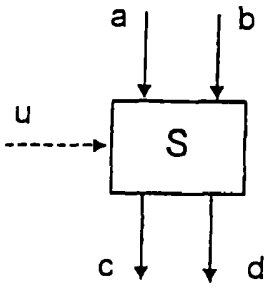
2. Spôsob podľa nároku 1, v y z n a č u j ú c i s a t ý m, že ako j-tá blokovo závislá prevodová operácia je použitá operácia permutácia bitov čiastkového kľúča v závislosti od spomenutého j-tého čiastkového bloku.

3. Spôsob podľa nároku 1, v y z n a č u j ú c i s a t ý m, že ako j-tá blokovo závislá prevodová operácia je použitá operácia cyklického odsadenia bitov čiastkového kľúča v závislosti od spomenutého j-tého čiastkového bloku.

4. Spôsob podľa nároku 1, v y z n a č u j ú c i s a t ý m, že ako j-tá blokovo závislá prevodová operácia je použitá substitučná operácia vykonaná na čiastkovom kľúči v závislosti od spomenutého j-tého čiastkového bloku.



Obr. 1.



Obr. 2.

u=1

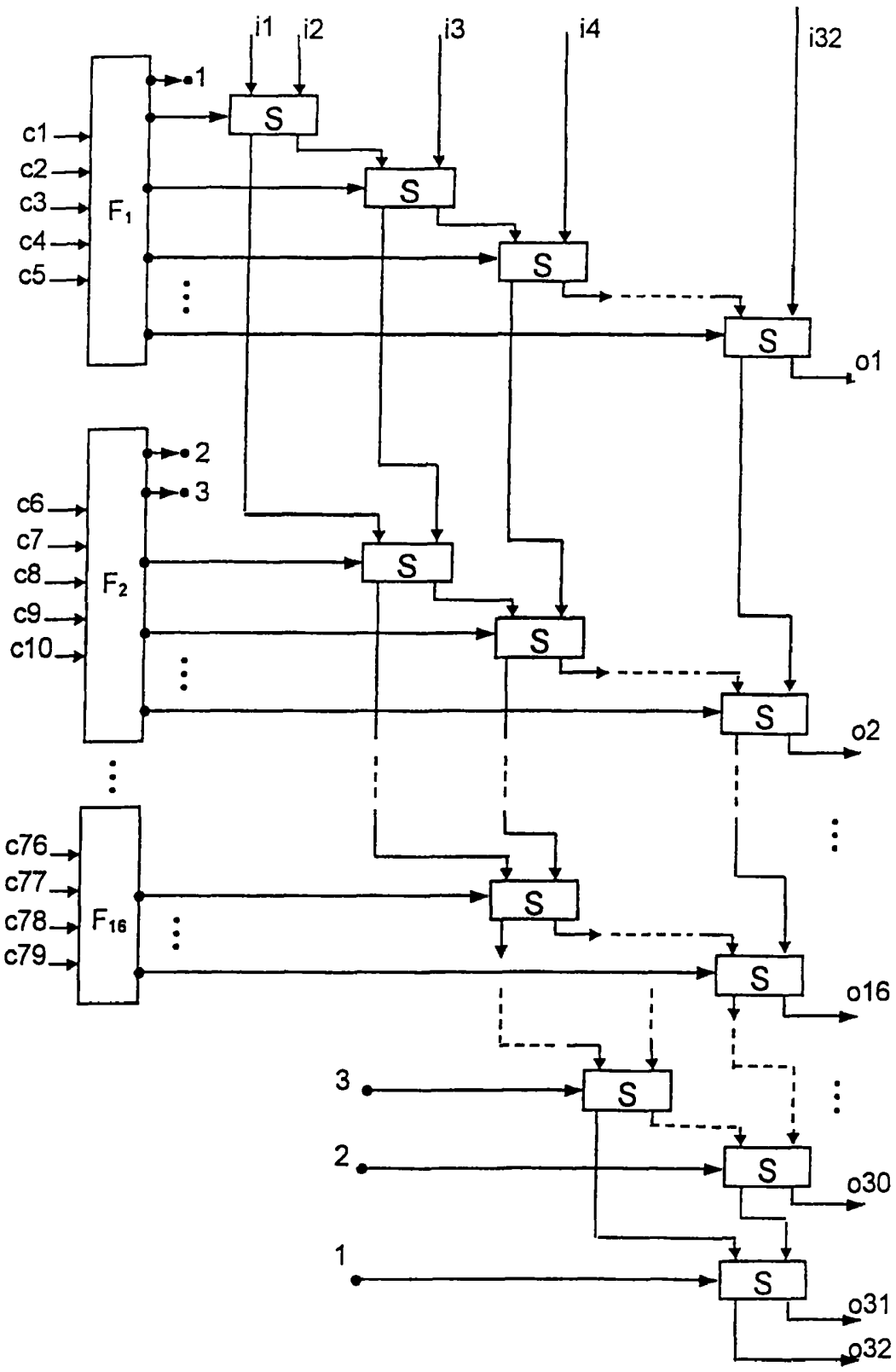
INPUT		OUTPUT	
a	b	c	d
1	0	1	0
0	1	0	1
0	0	0	0
1	1	1	1

Obr. 3.

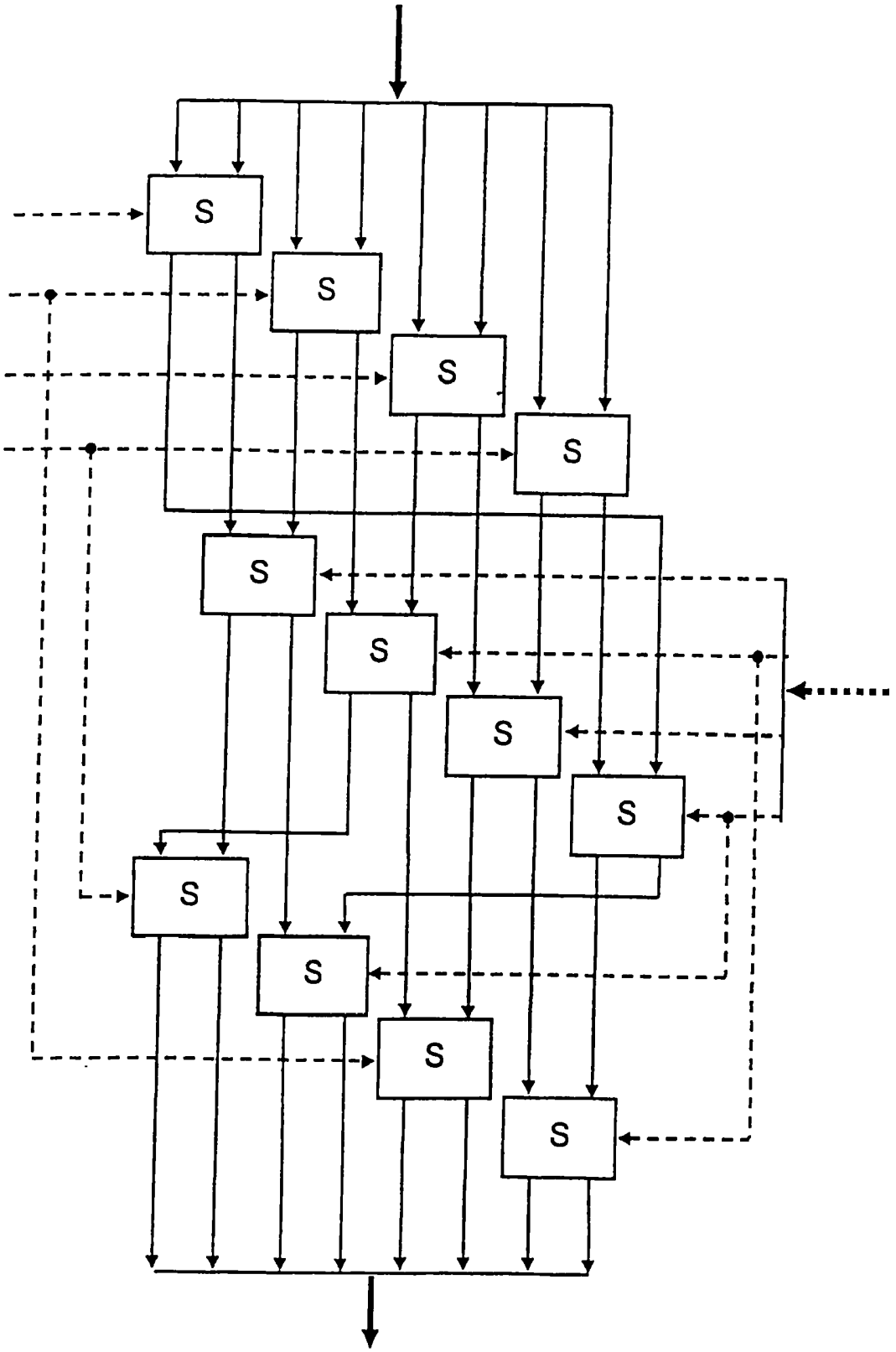
u=0

INPUT		OUTPUT	
a	b	c	d
0	1	1	0
1	0	0	1
0	0	0	0
1	1	1	1

Obr. 4.


i1
i2
i3
i4
i32
c1
c2
c3
c4
c5
F1
1
S
o1
c6
c7
c8
c9
c10
F2
2
3
o2
c76
c77
c78
c79
F16
o16
3
2
1
o30
o31
o32

Obr. 5.



Obr. 6.