PŘIHLÁŠKA VYNÁLEZU

zveřejněná podle § 31 zákona č. 527/1990 Sb.

(21) Číslo dokumentu:
2000 -3055

(13) Druh dokumentu: **A3**

(51) Int. Cl.[7]:
H 04 L 9/00

(19)
ČESKÁ
REPUBLIKA

ÚŘAD
PRŮMYSLOVÉHO
VLASTNICTVÍ

(22) Přihlášeno: **19.06.1998**

(32) Datum podání prioritní přihlášky: **24.02.1998 20.03.1998 22.04.1998**

(31) Číslo prioritní přihlášky: **1998/98103646 1998/98104851 1998/98107784**

(33) Země priority: **RU RU RU**

(40) Datum zveřejnění přihlášky vynálezu: **16.05.2001**
(Věstník č. 5/2001)

(86) PCT číslo: **PCT/RU98/00181**

(87) PCT číslo zveřejnění: **WO99/44330**

(71) Přihlašovatel:
OTKRYTOYE AKTSIONERNOYE OBSCHESTVO "MOSKOVSKAYA GORODSKAYA TELEFONNAYA SET", Moskva, RU;
MOLDOVYAN Alexandr Andreevich, Vsevolozhsk, RU;
MOLDOVYAN Nikolay Andreevich, Vsevolozhsk, RU;
SAVLUKOV Nikolay Viktorovich, Moskva, RU;

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

(74) Zástupce:
PATENTSERVIS PRAHA a.s., Jivenská 1, Praha 4, 14000;

(54) Název přihlášky vynálezu:

Způsob blokového kódování diskrétních dat

(57) Anotace:

Představované řešení patří do oblasti elektronické komunikace a počítačové technologie, a přesněji se týká kryptografických způsobů a zařízení pro kódování digitálních dat. Způsob obsahuje vytvoření kódovacího klíče formou sady dílčích klíčů, rozdělení datového bloku na určitý počet dílčích bloků N≥2, a střídavé převádění dílčích bloků vykonáváním dvojitých operací na dílčím bloku a dílčím klíči. Před vykonáním dvojité operace na i-tém dílčím bloku a dílčím klíči je provedena převodní operace na dílčím klíči v závislosti na j-tém dílčím bloku , kde j≠i. Převodní operaci závislou na j-tém dílčím bloku je permutační operace na bitech dílčího klíče závislá na j-tém dílčím bloku. Ve výhodném provedení je převodní operací závislou na j-tém dílčím bloku cyklická odsazovací operace na bitech dílčího klíče závislá na j-tém dílčím bloku. Dále je převodní operací závislou na j-tém dílčím bloku substituční operace na bitech dílčího klíče podle j-tého dílčího bloku.

CZ 2000 - 3055 A3

Způsob blokového kódování diskrétních dat

Oblast techniky

Představovaný vynález se týká oblasti elektronických komunikací a počítačových technologií, a přesněji se týká způsobů kódování pro kódování zpráv (informací).

Dosavadní stav techniky

V souhrnných vlastnostech nárokovaného způsobu jsou používány následující pojmy:

- tajný klíč představuje kombinaci bitů známou pouze legitimnímu uživateli;

- kódovací klíč je bitová kombinace používaná v kódovacích datových informačních signálech; kódovací klíč je kódovací proměnný prvek a je používán pro převádění dané zprávy nebo dané skupiny zpráv; kódovací klíč je vytvořen podle určených procedur a tajného klíče; při určitém počtu šifer je používán tento tajný klíč;

- šifra je souhrn elementárních kroků převodu vstupních dat za použití kódovacího klíče; šifra může být realizována jako počítačový program nebo jako samostatné elektronické zařízení;

- dílčí klíč je součástí kódovacího klíče používanou při jednotlivých elementárních kódovacích krocích;

- šifrování je proces realizování určitého způsobu převodu dat využívajícího kódovací klíč pro převádění dat do kryptogramu, který představuje pseudo-náhodnou sekvenci znaků, ze které je prakticky nemožné získat informace bez znalosti hodnoty kódovacího klíče;

- dešifrování je reverzní proces vzhledem k šifrovací proceduře; dešifrování zajišťuje obnovení informace podle kryptogramu, je-li znám kódovací klíč;

- kryptografická odolnost je míra bezpečnosti ochrany dat a reprezentuje intenzitu práce měřenou v počtu elementárních operací, které musí být vykonány za účelem obnovení informace z kryptogramu je-li znám převodní algoritmus, ovšem bez znalosti kódovacího klíče.

Jsou známy způsoby blokového datového kódování, viz např. šifra RC5 [R.Rivest, The RC5 Encryption Algorithm, Fast Software Encryption, Second International Workshop Proceedings (Leuven, Belgie, Prosinec 14-16, 1994), Lecture Notes in Computer Science, v.1008, Springer-Verlag, 1995, str.86-96]. Ve známém způsobu je datové blokové kódování provedeno vygenerováním kódovacího klíče ve tvaru skupiny dílčích klíčů, rozdělením převáděného datového bloku na dílčí bloky a střídavou změnou uvedeného použitím cyklické odsazovací operace, modulo 2 sčítací operace vykonané na dvou dílčích blocích, a modulo 2^{32} sčítací operace vykonané na dílčím bloku a dílčím klíči. V tomto případě jsou dílčí klíče použity podle pevného programu, t.j. v daném kroku vykonávání binární operace mezi dílčím blokem a dílčím klíčem hodnota dílčího klíče nezávisí na datovém vstupním bloku. Tento způsob blokového kódování zajišťuje vysokou kódovací rychlost, je-li realizován jako počítačový program.

Tento způsob však selhává při zajištění dostatečné odolnosti proti diferenciální a lineární 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], což je způsobeno díky tomu, že v tomto způsobu jsou používány v daných kódovacích krocích pevné dílčí klíče pro všechny možné vstupní bloky.

Nárokovanému blokovému kódovacímu způsobu je ve své technické podstatě nejbližším způsob popsaný v US standardu DES [National Bureau of Standards. Data Encryption Standard. Federal Information Proceedings Standard

Publication 46, January 1977]. Tento způsob obsahuje vygenerování kódovacího klíče ve tvaru sady 48-bitových dílčích klíčů, rozdělení vstupního bloku diskrétních dat na dva 32-bitové dílčí bloky L a R, a střídavé převádění dílčích bloků při řízení tajným klíčem. Celkově je vykonáno 16 cyklů převodu 32-bitového dílčího bloku. Každý cyklus převodu dílčího bloku je uskutečněn vykonáním následujících procedur: (1) rozšíření dílčího bloku R na 48 bitů opakováním určitých bitů tohoto dílčího bloku: R->R´, (2) provedení modulo 2 sčítací operace na dílčím bloku a dílčím klíči, (3) rozdělení dílčího bloku R´ na osm 6-bitových dílčích bloků, (4) vykonání substituční operace na každém 6-bitovém dílčím bloku nahrazením 6-bitových dílčích bloků 4-bitovými dílčími bloky podle známých substitučních tabulek, (5) zkombinování osmi 4-bitových dílčích bloků do 32-bitového dílčího bloku 2, (6) vykonání operace bitové permutace R dílčích bloků podle určeného pravidla, (7) vykonání modulo 2 sčítací operace na dílčím bloku R s dílčím blokem L. Při vykonávání aktuálního kódovacího cyklu je používán pevný dílčí klíč pro všechny možné datové vstupní bloky. Dílčí klíče používané při převádění dílčích bloků jsou vygenerovány pod řízením 56-bitového tajného klíče. Tento způsob informačního blokového kódování dosahuje velké převodní rychlosti, je-li realizován pomocí speciálního elektronického obvodu.

Tento způsob má však některé nevýhody, jmenovitě dosahuje nízké kódovací rychlosti, je-li realizován pomocí software. Dále, tento způsob využívá 56-bitového tajného klíče, což umožňuje výkonným moderním počítačům odhalit tajný klíč výběrem možných klíčových hodnot. Toto vyžaduje vykonání několika kódovacích procedur využívajících různé tajné klíče, což činí způsob obtížným pro získání vysoké kódovací rychlosti dokonce i v případě hardwarové realizace.

Základem vynálezu je úkol vyvinout způsob blokového kódování diskrétních dat, kde převod datového dílčího bloku bude ovlivněn tak, aby byl snížen počet převodních operací uvažovaných pro jeden vstupní datový bit, přičemž bude zároveň poskytovat vysokou kryptografickou odolnost, což povede ke zvýšené kódovací rychlosti.

Podstata vynálezu

Výše zmíněného cíle je dosaženo faktem, že způsob blokového kódování diskrétních dat zahrnuje vygenerování kódovacího klíče jako sadu dílčích klíčů, rozdělení datového bloku na N>=2 dílčích bloků a střídavé převádění dílčího bloku vykonáním dvojité operace na dílčím bloku a dílčím klíči, přičemž novou vlastností podle vynálezu je vykonávání j-té blokově-závislé dílčí operace, kde j≠1, na dílčím klíči před vykonáním dvojité operace na i-tém dílčím bloku a dílčím klíči.

Díky tomuto řešení závisí struktura dílčího klíče používaná v daném kódovacím kroku na převáděných datech a tak jsou v daném převodním kroku použity odlišné modifikované hodnoty dílčího klíče pro odlišné vstupní bloky, díky čemuž je poskytována vysoká kryptografická odolnost proti diferenciální kryptoanalýze, přičemž je zároveň snížen počet kódovacích cyklů a to vede ke zvýšení rychlosti kryptografického převodu.

Novou vlastností je také fakt, že jako j-tá blokově-závislá převodní operace je využita bitová permutační operace j-tého blokově-závislého dílčího klíče.

Díky tomuto řešení je poskytována zvýšená kódovací rychlost, je-li nárokovaný způsob realizován formou elektronického kódovacího zařízení.

Novou vlastností je také to, že bitová cyklická odsazovací operace j-tého blokově-závislého dílčího klíče je použita jako j-tá blokově závislá převodní operace.

Díky tomuto řešení je poskytována zvýšená kódovací rychlost, je-li nárokovaný způsob realizován jako počítačový kódovací software.

Další novou vlastností je to, že j-tá blokově závislá permutační operace s dílčím klíčem je využita jako j-tá blokově závislá převodní operace.

Díky tomuto řešení je poskytováno dodatečné zvýšení kódovací kryptografické odolnosti, zároveň zajišťující vysokou kódovací rychlost, je-li nárokovaný způsob realizován jako počítačový kódovací software.

Níže bude podstata nárokovaného vynálezu popsána detailněji na provedeních s odkazy na připojené obrázky.

Přehled obrázků na výkresech

Obr.1 představuje zobecněný kódovací diagram podle nárokovaného způsobu.

Obr.2 představuje blokový diagram elementárně řízeného spínače, kterým je základní prvek řízeného permutačního bloku. Je-li u=1, vstupní bity nejsou permutovány, t.j. výstupní signály se shodují se vstupními signály. Je-li u=0, pak jsou vstupní bity permutovány.

Obr.3 představuje tabulku vstupních a výstupních signálů elementárně řízeného spínače, má-li potenciál řídícího signálu vysokou hodnotu (high).

Obr.4 představuje tabulku vstupních a výstupních signálů elementárně řízeného spínače, má-li potenciál řídícího signálu nízkou hodnotu (low).

Obr.5 představuje schematicky strukturu řízeného permutačního bloku, skládajícího se ze sady bloků stejného typu, elementárních spínačů, který realizuje 2^{79} různých

permutací vstupních bitů v závislosti na hodnotě 79-bitového řídícího kódu.

Obr.6 představuje diagram zjednodušeného řízeného permutačního bloku.

Příklady provedení vynálezu

Vynález je vysvětlen pomocí zobecněného diagramu kryptografického převodu datového bloku založeného na nárokovaném způsobu, který je představován na obr.1, kde P je blok řízené operace vykonané na dílčím klíči; A a B jsou převedené n-bitové dílčí bloky; K_{2r}, K_{2r-1} jsou m-bitové dílčí klíče (obecně m≠n); Q(2r), Q(2r-1) jsou g-bitové dodatečné dílčí klíče; znaménko „⊕" označuje modulo 2 bit-po-bitu sčítací operaci; znaménko „⊗" označuje modulo 2^n sčítací operaci. Tučné nepřerušované čáry označují n-bitovou signálovou přenosovou sběrnici, slabé tečkované čáry označují přenos jednoho řídícího bitu. Tučné tečkované čáry označují sběrnici pro přenos n řídících signálů, s nimiž jsou použity převáděné bity dílčího bloku. Tučné tečkované čáry označují také sběrnici pro přenos h bitů dodatečných dílčích klíčů Q(2r) a Q(2r-1), které slouží ke změně operace v závislosti na dílčím bloku, který je převáděn. V konkrétních případech nemusí být použity dodatečné dílčí klíče.

Obr.1 ukazuje jednotlivý (r-tý) kódovací cyklus. V závislosti na přesném typu použité řídící operace a na požadované převodní rychlosti může být nastaveno od 6 do 10 nebo více cyklů. Jednotlivý převodní cyklus obsahuje vykonání následující sekvence procedur:

(1) převedení dílčího klíče K_{2r} v závislosti na hodnotě dílčího bloku A a na hodnotě dodatečného dílčího

klíče $\underline{Q(2r)}$, jehož výsledkem je vygenerování převedené hodnoty dílčího klíče $\underline{P_{A,Q(2r)}(K_{2r})}$ na výstupu bloku $\underline{P_1}$;

(2) převedení dílčího bloku $\underline{B}$ vykonáním modulo 2 bit-po-bitu sčítací operace na hodnotě $\underline{P_{A,Q(2r)}(K_{2r})}$ a dílčího bloku $\underline{B}$: $B:=B\oplus P_{A,Q(2r)}(K_{2r})$, kde znaménko „:=" označuje přiřazovací operaci;

(3) převedení dílčího bloku $\underline{A}$ vykonáním modulo 2^n sčítací operace na dílčím bloku $\underline{A}$ a dílčím bloku $\underline{B}$: $A:=A\otimes B$;

(4) převedení dílčího klíče $\underline{K_{2R-1}}$ v závislosti na hodnotě dílčího bloku $\underline{B}$ a na hodnotě dodatečného dílčího klíče $\underline{Q(2r-1)}$ jehož výsledkem je vygenerování převedené hodnoty dílčího klíče $\underline{P_{A,Q(2r-1)}(K_{2r-1})}$ na výstupu bloku $\underline{P_2}$;

(5) převedení dílčího bloku $\underline{A}$: $A:=A\oplus P_{A,Q(2r-1)}(K_{2r-1})$;

(6) převedení dílčího bloku $\underline{B}$: $B:=B\otimes A$.

V závislosti na konkrétním provedení navrhovaného způsobu blokového kódování diskrétních informací může být při provádění jednotlivých kódovacích cyklů použit stejný pár m-bitových dílčích klíčů $\underline{K_2}$ a $\underline{K_1}$ (dodatečné g-bitové dílčí klíče $\underline{Q(2)}$ a $\underline{Q(1)}$). Je možné provedení, kdy jsou v jednotlivých cyklech použity nezávislé dílčí klíče $\underline{K_{2r}}$ a $\underline{K_{2r-1}}$($\underline{Q(2r)}$ a $\underline{Q(2r-1)}$). Je-li například počet cyklů r=3, první cyklus používá dílčí klíče $\underline{K_2}$ a $\underline{K_1}$ ($\underline{Q(2)}$ a $\underline{Q(1)}$), druhý cyklus používá dílčí klíče $\underline{K_4}$ a $\underline{K_3}$ ($\underline{Q(4)}$ a $\underline{Q(3)}$), třetí cyklus používá dílčí klíče $\underline{K_6}$ a $\underline{K_5}$ ($\underline{Q(6)}$ a $\underline{Q(5)}$). Dílčí klíče $\underline{K_{2r}}$, $\underline{K_{2r-1}}$ a dodatečné dílčí klíče $\underline{Q(2r)}$ a $\underline{Q(2r-1)}$ mohou být vytvořeny podle speciálních procedur v závislosti na tajném klíči. Je možné provedení, ve kterém jsou dílčí klíče $\underline{K_{2r}}$, $\underline{K_{2r-1}}$ a dodatečné dílčí klíče $\underline{Q(2r)}$ a $\underline{Q(2r-1)}$ vytvořeny vygenerováním náhodných pravidel.

Možné technické provedení nárokovaného způsobu je vysvětleno pomocí jeho následujících specifických provedení.

Příklad 1.

Tento příklad vysvětluje kódování 64-bitových datových bloků využívající řízené permutace jako operace vykonané na dílčím klíči v závislosti na jednom z bloků, který je převáděn. Kódovací klíč je vygenerován jako 16 dílčích klíčů $\underline{K}_1$, $\underline{K}_2$, $\underline{K}_3$ ...$\underline{K}_{16}$, z nichž každý má délku 32 bitů. Dodatečné dílčí klíče nejsou zavedeny. Datový vstupní blok je rozdělen na dva 32-bitové dílčí bloky $\underline{A}$ a $\underline{B}$. Kódování vstupního bloku je popsáno následujícím algoritmem:

1. Nastavení čísla cyklu:

r:=1.

2.Převedení dílčího bloku $\underline{B}$ podle výrazu:

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

kde $\underline{P_A(K_{2r})}$ označuje operaci permutace bitů dílčího klíče $\underline{K}_{2r}$ vykonanou v závislosti na hodnotě dílčího bloku $\underline{A}$.

3.Převedení dílčího bloku $\underline{A}$ podle výrazu:

$A:=A \otimes B$.

4.Převedení dílčího bloku $\underline{A}$ podle výrazu:

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

kde $\underline{P_B(K_{2r-1})}$ označuje operaci permutace bitů dílčího klíče $\underline{K}_{2r-1}$ vykonanou v závislosti na hodnotě dílčího bloku $\underline{B}$.

5.Převedení dílčího bloku $\underline{B}$ podle výrazu:

$B:=B \otimes A$.

6.Jestliže $r \neq 8$, přičti jedničku k číslu r:=r+1 a přejdi na krok 2, jinak STOP.

Tento algoritmus je orientován pro realizaci formou elektronického obvodu. Operace bitových permutací dílčích klíčů v závislosti na jednom z převáděných dílčích bloků může být vykonána použitím řízeného permutačního bloku realizovaného na základě použití sady elementárních spínačů, které vykonávají operaci permutace dvou bitů.

Obr.2 vysvětluje operaci elementárního spínače, kde u je řídící signál, a a b jsou datové vstupní signály, c a d jsou datové výstupní signály.

Tabulky na obr.3 a obr.4 ukazují závislost výstupního signálu na vstupním signálu a řídících signálech. Z těchto tabulek bude zřejmé, že při u=1 se řada a mění s řadou c a řada b s řadou d. Je-li u=0, pak se řada a mění s řadou d a řada b s řadou d. Proto je-li řídící signál roven jedné, žádné dva vstupní bity nejsou permutovány, zatímco je-li řídící signál roven nule, vstupní bity jsou permutovány.

Obr.5 ukazuje možné provedení řízeného permutačního bloku využívajícího sadu elementárních spínačů S. Tento příklad odpovídá bloku P obsahujícímu 32-bitový informační vstup a 79-bitový řídící vstup. Bity aktuálně převáděného dílčího klíče jsou použity jako informační signály. 32 bitů jednoho z dílčích bloků a 47 bitů jednoho z dodatečných dílčích klíčů je použito jako řídící signály.

Počet možných verzí permutačních operací je roven počtu možných kódových kombinací na řídícím vstupu a pro blok P se strukturou zobrazenou na obr.2 činí 2^{79}. Tento řízený permutační blok realizuje jedinečnou permutaci vstupních bitů pro každou možnou hodnotu kódové kombinace na řídícím vstupu, jejichž počet je 2^{79}. Další informační vstupy řízeného permutačního bloku jsou označeny i1, i2,..,i32, vnější výstupy jsou označeny o1,o2,..,o32, řídící vstupy jsou označeny c1,c2,..,c79. Elementární spínače S jsou spojeny takovým způsobem, aby vytvořily pole sestávající z 31 řad. V první řadě je zapojeno 31 elementárních spínačů S, v druhé řadě 30 spínačů, ve třetí řadě 29 spínačů, atd. V každé následující řadě je počet elementárních spínačů snížen o 1. V nejnižší 31.řadě je zapojen jeden elementární spínač.

Řada označená j≠31 má 33-j vstupů, 33-j výstupů a 32-j řídících vstupů. Poslední (nejvíce napravo) vstup j-té řady je vnější výstup řízeného permutačního bloku, zbylých 32-j výstupů j-té řady je připojeno k odpovídajícím vstupům (j+1)-té řady. Poslední 31.řada má dva výstupy a oba z nich jsou vnější výstupy řízeného permutačního bloku. Pouze na jeden řídící vstup každé řady je aplikován jednotkový (u=1) řídící signál. Pro splnění těchto požadavků jsou poskytovány dva-třicet-dva-stupňové dekódovače $\underline{F}_1$, $\underline{F}_2$,..,$\underline{F}_{15}$ a dva-šestnáct-stupňový dekódovač $\underline{F}_{16}$. Dekódovače $\underline{F}_1$, $\underline{F}_2$,..,$\underline{F}_{15}$ mají pět vnějších řídících vstupů, ke kterým je přiveden náhodný 5-ti bitový binární kód, a 32 výstupů. Tyto dekódovače generují pouze na jednom výstupu jednotkový signál. Na zbývajících 31 výstupech je nastaven nulový signál. Dekódovač $\underline{F}_{16}$ má 4 vstupy, ke kterým je přiváděn libovolný 4-bitový binární kód, a 16 výstupů, z nichž pouze na jednom je nastavena hodnota signálu jedna. Pro všechny dekódovače $\underline{F}_1$, $\underline{F}_2$,..,$\underline{F}_{15}$ a $\underline{F}_{16}$ každá vstupní binární kódová hodnota nastavuje jedinečné možné výstupní číslo, při kterém je nastaven jednotkový signál (u=1).

Část výstupů kódovače F_h, kde h≤15, je připojena k řídícím vstupům řady očíslované h (32-h výstupů), zatímco část výstupů je připojena k řídícím vstupům (32-h)-té řady (h výstupů). Proto je v každé řadě nastaven řídící signál u=1 pouze na jednom elementárním spínači. Řadový vstup spojený s pravým vstupem elementárního spínače, na který je aplikován jednotkový řídící signál, komutuje s vnějším výstupem řízeného permutačního bloku odpovídajícího dané řadě. Je-li jednotkový řídící signál aplikován na elementární spínač umístěný nejvíce nalevo, vnější výstup řízeného permutačního bloku (blok) komutuje s řadovým vstupem umístěným nejvíce nalevo. První řada komutuje jeden z vnějších výstupů $\underline{i1}$,$\underline{i2}$,..,$\underline{i32}$ bloku $\underline{P}$ s vnějším výstupem $\underline{o1}$ a zbývajících 31 vnějších vstupů se vstupy druhé řady.

Druhá řada komutuje jeden ze zbývajících 31 vnějších vstupů s vnějším výstupem o2 a zbývajících 30 vnějších vstupů se vstupy třetí řady, atd. Tato struktura bloku P implementuje jedinečnou permutaci vstupních bitů pro všechny hodnoty binárního kódu přivedené na 79-bitový řídící vstup bloku P.

Je proveditelná následující verze použití řídícího permutačního bloku P s 32-bitovým informačním vstupem a 79-bitovým řídícím vstupem. 32 bitů dílčího bloku A a 47 bitů dodatečného 47-bitového dílčího klíče Q(2r) může být použito jako řídící signály aplikované na 79-bitový řídící vstup řízeného permutačního bloku P. V tomto případě, v závislosti na 47-bitovém dodatečném dílčím klíči, je vytvořena jedna z 2^{47} různých modifikací bitové permutační operace, která závisí na hodnotě vstupního bloku. Přitom každá modifikace této operace zahrnuje 2^{32} různých operací permutace bitů dílčího klíče K_{2r}, kde výběr určité permutační operace je určen hodnotou dílčího bloku A. Výběr modifikace není předdefinován, protože je definován dodatečným dílčím klíčem Q(2r), který je přímo prvkem tajného klíče nebo je závislý na tajném klíči. Toto dále zvyšuje odolnost proti kryptografickému převodu. Jestliže kódovací zařízení používá dva bloky P mající strukturu zobrazenou na obr.2, počet možných modifikovaných kombinací řízené permutační operace nastavené na blocích P v závislosti na dodatečných 47-bitových dílčích klíčích může být nastaven až do $(2^{47})^2=2^{94}$, je-li použit tajný klíč o délce 94 bitů.

Moderní technologie výroby integrovaných obvodů umožňuje díky jednoduché struktuře bloků P snadnou výrobu kryptografických mikroprocesorů obsahujících řízené permutační bloky se vstupní kapacitou 32 a 64 bitů a poskytujících kódovací rychlost až do 1 Gbit/s a výše.

Obr.6, na kterém tenké plné čáry označují přenos jednoho bitu dílčího klíče, ukazuje možnou realizaci řízeného permutačního bloku využívajícího sadu

elementárních spínačů S. Tento příklad řízeného permutačního bloku se shoduje s řízeným permutačním blokem. Obsahuje 8-bitový vstup pro informační signály (bity dílčího klíče) a 8-bitový vstup pro řídící signály (bity datového dílčího bloku označené tečkovanými čárami podobnými čárám na obr.1). Podobným způsobem je možné vytvořit libovolný řízený permutační blok, obsahující například 64-bitový vstup pro informační signály a 128-bitový vstup pro řídící signály. Při používání řízeného permutačního bloku s 32-bitovým informačním vstupem je počet různých permutací roven 2^{32}. To znamená, že při kódování dvou různých datových bloků je možnost opakování určité permutace v dané sadě rovna 2^{-32}, zatímco možnost opakování permutací v z sadových krocích je rovna 2^{-32z}. Sada pozměněných hodnot dílčího klíče použitá pro převod každé vstupní zprávy je proto prakticky jedinečná, což zajišťuje vysokou kryptografickou odolnost kódování.

Při použití zjednodušené struktury řízeného permutačního bloku zobrazeného na obr.6 je jednoduché vyrobit kryptografické mikroprocesory obsahující řízené permutační bloky se vstupní kapacitou až do 128 bitů. Použití řízených permutačních operací na 128-bitovém dílčím klíči umožňuje získat vyšší kryptografickou odolnost kódování. Řízený permutační blok je kombinační elektrický obvod, který poskytuje vysokou rychlost vykonávání řízených permutací.

Příklad 2.

Tento příklad vysvětluje použití cyklické odsazovací operace závislé na dílčích blocích, které jsou převáděny, a prováděné na dílčích klíčích. Kódovací klíč je vygenerován ve tvaru 16 dílčích klíčů $\underline{K}_1, \underline{K}_2, \underline{K}_3, .., \underline{K}_{32}$, z nichž každý má délku 32 bitů. Vstupní 64-bitový datový blok je rozdělen na dva 32-bitové dílčí bloky A a B. Kódování vstupního bloku je popsáno následujícím algoritmem:

1.Nastavení čísla cyklu:

r:=1.

2.Převedení dílčího bloku B podle výrazu:

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

kde $K_{2r}<<<A$ označuje operaci cyklického odsazení nalevo pomocí A bitů vykonanou na dílčím klíči K_{2r}.

3.Převedení dílčího bloku A podle výrazu:

$A:=A \otimes B$,

kde „⊗" je modulo 2^{32} sčítací operace.

4.Převedení dílčího bloku A podle výrazu:

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

kde $K_{2r-1}<<<B$ označuje operaci cyklického odsazení nalevo pomocí B bitů vykonanou na dílčím klíči K_{2r-1}.

5.Převedení dílčího bloku B podle výrazu:

$B:=B \otimes A$.

6.Jestliže $r \neq 16$, přičti jedničku k číslu r:=r+1 a přejdi na krok 2, jinak STOP.

Logický vzor převodního cyklu je vysvětlen na obr.1, kde bloky P_1 a P_2 v tomto příkladu představují operační blok vykonávající operaci cyklického odsazování bitů odpovídajícího dílčího klíče v závislosti na dílčích blocích, které jsou převáděny. Tento algoritmus je orientován pro realizaci formou počítačového programu. Moderní mikroprocesor rychle vykonává cyklické odsazovací operace v závislosti na hodnotě proměnné uložené v jednom z registrů. Díky tomuto faktu poskytuje popsaný algoritmus, je-li realizován softwarově, kódovací rychlost okolo 40 Mbit/s pro hromadně-rozšířený mikroprocesor Pentium/200. Je-li nastaveno 10 kódovacích cyklů, je dosaženo rychlosti okolo 60 Mbit/s.

Příklad 3.

Tento příklad vysvětluje použití substitučních operací závislých na dílčích blocích, které jsou převáděny, a vykonávaných na dílčích klíčích. V představovaném

příkladu představují bloky $\underline{P}_1$ a $\underline{P}_2$ operační blok vykonávající substituční operaci v závislosti na odpovídajících dílčích blocích. Substituční operací je rozuměna operace nahrazení binární hodnoty signálu na vstupu operačního bloku $\underline{P}$ jinou binární hodnotou (nastavenou na výstupu operačního bloku), která je vybrána v závislosti na hodnotě na vstupu bloku $\underline{P}$ v souladu s určitou vstupní tabulkou. Mohou být realizovány dvě substituční verze:

(1) n-bitový vstupní binární vektor je nahrazen n-bitovým výstupním binárním vektorem, kde různým vstupním binárním vektorům odpovídají různé výstupní binární vektory;

(2) m-bitový binární vektor je nahrazen n-bitovým binárním vektorem, kde $n \neq m$, přičemž různým vstupním binárním vektorům mohou odpovídat jak rozdílné tak i shodné výstupní binární vektory.

Vysvětleme si nyní specifikující závislost prvního typu substituční operace na dílčím bloku dat, který je převáděn. Předpokládejme, že substituční operace jsou vykonávány na binárních vektorech s n-bitovou délkou, kde n je celé číslo, pro zajištění substituční operace o kapacitě nxn (označení nxn označuje, že vstupem pro substituční operaci je binární vektor o délce n bitů, a výstupní binární vektor má rovněž délku n bitů). Je vyžadováno použití tabulky obsahující dvě řady číslic:

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

kde $N=2^n$. Ve spodní řadě této tabulky se nacházejí všechny možné hodnoty n-bitového bloku odpovídající jednou, avšak v libovolném pořadí. Správné pořadí umístění čísel ve spodní řadě určuje specifickou verzi substituční tabulky a proto také specifickou verzi substituční operace vykonávané použitím této tabulky. Vykonávání substituční

operace je provedeno následovně. V horní řadě je vybráno číslo rovné hodnotě vstupního bloku. Hodnota nacházející se pod tímto číslem ve spodní řadě je použita jako výstupní blok. Substituční tabulka proto může být umístěna v pracovní paměti počítače jako po sobě jdoucí soustava n-bitových počítačových slov umístěných uvnitř buněk s adresami $\underline{w}_0,\underline{w}_1,\underline{w}_2,..,\underline{w}_{N-1}$. V tomto případě slouží hodnota vstupního binárního vektoru $\underline{Y}$ pro výpočet adresy w_0+Aby slova, které je vybráno jako výstup binárního vektoru. Tento způsob reprezentace substituční tabulky vyžaduje použití paměťové kapacity rovné $Nn=2^n n$ bitů. Uvažujme počet substitučních tabulek roven 2^L (požadovaná kapacita paměti bude v tomto případě $2^L Nn$ bitů) a umístěme substituční tabulky nepřerušovaně jednu za druhou. Vezměme hodnotu adresy $\underline{w}_0$ z prvního bitu slova tabulky jako tabulkovou adresu s číslem $\underline{v}$. Nechť tabulková adresa s číslem v=0 je $\underline{s}$. V tomto případě substituční tabulková adresa s libovolným číslem $\underline{v}$ je s+vN. Je-li řídící binární vektor specifikován tak, že určuje číslo aktuální substituční tabulky stejně jako aktuální vstupní binární vektor, pak je substituční operace provedena nahrazením aktuálního vstupního bloku n-bitovým slovem umístěným na adrese s+vN+Y, kde $\underline{Y}$ je hodnota vstupního binárního vektoru, na kterém je vykonávána aktuální substituční operace. Použitím tohoto vztahu je jednoduché specifikovat výběr substituční tabulky s číslem $\underline{v}$ a vykonat substituci na vstupním binárním vektoru s hodnotou $\underline{Y}$. Vzhledem k tomu je specifikování závislosti substitučních tabulek na hodnotě řídícího binárního vektoru a vykonávání substituční operace provedeno mikroprocesorem velmi rychle, jsou-li vybrány odpovídající hodnoty parametrů $\underline{L}$ a $\underline{n}$, například je-li L=5 a n=8. Jsou-li tyto parametry vybrány, pak je za účelem lokalizace substituční tabulky vyžadováno 8kbytů pracovní paměti, což je přijatelné, protože moderní počítače mají

kapacitu pracovní paměti o mnoho řádů vyšší než je tato hodnota (od 1 do 64 Mbytů a více).

Nyní vysvětlíme specifikující závislost druhého typu substituční operace na datovém dílčím bloku pomocí příkladu 16x32 substitucí specifikovaných použitím číslované sekvence 32-bitových binárních vektorů $\underline{X}_j, J=0,1,2,..,12^{16}-1$. Předpokládá se, že sekvence $\underline{X}_j$ je známá a týká se popisu kódovacího algoritmu. Substituční operace na 16-bitovém klíči $\underline{k}$ je provedena v závislosti na převáděném dílčím bloku $\underline{b}$ následovně:

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

(2) 16-bitový binární vektor $\underline{k}$ je nahrazen 32-bitovým binárním vektorem $\underline{X}_j$;

Kódování 64-bitových datových bloků založené na substitučních operacích využívajících sekvenci 32-bitových binárních vektorů $\underline{X}_j (J=0,1,2,..,12^{16}-1)$ na dílčích klíčích závislých na datových dílčích blocích, které jsou převáděny, může být uskutečněno například následovně. Kódovací klíč je vygenerován formou 16 dílčích klíčů $\underline{K}_1, \underline{K}_2, \underline{K}_3, .., \underline{K}_{32}$, z nichž každý má délku 16 bitů. Vstupní datový blok je rozdělen na dva 32-bitové dílčí bloky $A=a_2|a_1$ a $B=b_2|b_1$ reprezentovaných jako kaskáda příslušných 16-bitových dílčích bloků $\underline{a}_1, \underline{a}_2$ a $\underline{b}_1, \underline{b}_2$. Kódování vstupního bloku je popsáno následujícím algoritmem:

1.Nastavení čísla cyklu r=1.

2.Převedení dílčího bloku $\underline{B}$ podle výrazu:

$B:=B \oplus F(K_{4r}, a_1)$,

kde $\underline{F}(K_{4r}, a_1)$ označuje substituční operaci na dílčím klíči $\underline{K}_{4r}$ v závislosti na dílčím bloku $\underline{a}_1$.

3.Převedení dílčího bloku $\underline{A}$ podle výrazu:

$A:=A+ B (\bmod 2^{32})$.

4.Převedení dílčího bloku $\underline{A}$ podle výrazu:

$A:=A \oplus F(K_{4r-1}, b_1)$,

kde $\underline{F}(K_{4r-1},b_1)$ označuje substituční operaci na dílčím klíči $\underline{K}_{4r-1}$ vykonanou v závislosti na dílčím bloku $\underline{b}_1$.

5.Převedení dílčího bloku $\underline{B}$ podle výrazu:

$B:=B+ A(mod2^{32})$.

6.Převedení dílčího bloku $\underline{B}$ podle výrazu:

$B:=B\oplus F(K_{4r-2},b_2)$.

7.Převedení dílčího bloku $\underline{A}$ podle výrazu:

$A:=A+ B(mod2^{32})$.

8.Převedení dílčího bloku $\underline{A}$ podle výrazu:

$A:=A\oplus F(K_{4r-3},b_2)$.

9. Převedení dílčího bloku $\underline{B}$ podle výrazu:

$B:=B+ A(mod2^{32})$.

10.Jestliže $r\neq4$, pak přičti jedničku k číslu $r:=r+1$ a přejdi na krok 2, jinak STOP.

Tento algoritmus používá známou substituční tabulku o velikosti 240 kbytů, která zabírá malou část kapacity pracovní paměti moderního počítače. Operace výběru binárních vektorů z pracovní paměti podle předdefinovaných adres je vykonána v malém počtu strojových cyklů, díky čemuž softwarové provedení navrhovaného způsobu blokového kódování se substitučními operacemi vykonávanými na dílčích klíčích v závislosti na převáděných dílčích blocích poskytuje kódovací rychlost od 20 do 60 Mbit/s (v závislosti na specifickém provedení) pro široce rozšířený mikroprocesor Pentium/200.

Průmyslová využitelnost

Uvedené příklady ukazují, že navrhovaný způsob blokového kódování diskrétních dat je technicky proveditelný a umožňuje řešení problému, který jsme definovali.

Diskutované příklady jsou snadno proveditelné, například formou specializovaných mikroelektronických kódovacích obvodů (Příklad 1) a formou kódovacího

počítačového software (Příklady 2 a 3) a zajišťují kódovací rychlost až do 1 Gbit/s a vyšší (Příklad 1), jedná-li se o hardwarovou realizaci, a až do 60 Mbit/s, jedná-li se o softwarovou realizaci a při použití široce rozšířeného mikroprocesoru Pentium/200 (Příklady 2 a 3).

P A T E N T O V É N Á R O K Y

1. Způsob blokového kódování diskrétních dat, obsahující vygenerování kódovacího klíče formou sady dílčích klíčů, rozdělení datového bloku na N≥2 dílčích bloků a střídavé převádění zmíněných dílčích bloků vykonáváním dvojité operace na dílčím bloku a dílčím klíči, vyznačující se tím, že před vykonáním zmíněné dvojité operace na i-tém dílčím bloku a dílčím klíči je provedena převodní operace na dílčím klíči v závislosti na j-tém dílčím bloku, kde j≠i.
2. Způsob podle nároku 1, vyznačující se tím, že jako j-tá blokově závislá převodní operace je použita operace permutace bitů dílčího klíče v závislosti na zmíněném j-tém dílčím bloku.
3. Způsob podle nároku 1, vyznačující se tím, že jako j-tá blokově závislá převodní operace je použita operace cyklického odsazení bitů dílčího klíče v závislosti na zmíněním j-tém dílčím bloku.
4. Způsob podle nároku 1, vyznačující se tím, že jako j-tá blokově závislá převodní operace je použita substituční operace provedená na dílčím klíči v závislosti na zmíněním j-tém dílčím bloku.