

# Crypto-World

Informační sešit GCUCMP

ISSN 1801-2140

Ročník 9, číslo 3/2007

15. březen 2007

## 3/2007

Připravil: Mgr. Pavel Vondruška

Sešit je přednostně distribuován registrovaným čtenářům.

Starší sešity jsou dostupné na adrese

<http://crypto-world.info>

(1240 registrovaných odběratelů)



### Obsah :

str.

A. O speciální blokové šifře DN a hašovací funkci HDN (T.Rosa)	2-3
B. Rodina speciálních blokových šifer DN a hašovacích funkcí nové generace HDN typu SNMAC (V.Klíma)	4-26
C. Najväčšia tma je pod lampou – STEGANOGRAFIA, část II. (R.Cinkais)	27-33
D. Šifrování v MS Office (P.Tesař)	34
E. O čem jsme psali v březnu 2000 - 2006	35-36
F. Závěrečné informace	37

## A. O speciální blokové šifře DN a hašovací funkci HDN Dr. Tomáš Rosa, kryptolog, eBanka a.s (trosa@ebanka.cz)

Na podzim minulého roku vzbudil pozornost návrh nové rodiny hašovacích funkcí typu SNMAC, u nichž Dr. Klíma prokázal mimořádné bezpečnostní vlastnosti, které současným hašovacím funkcím chybí. Tyto funkce byly založeny na tzv. speciálních blokových šifrách, které však v té době ještě byly utajeny v rámci projektu NBÚ. Vzhledem k významu těchto návrhů a možnosti jejich oponentury v mezinárodním konkurenčním prostředí, je NBÚ uvolnil ke zveřejnění. Dnes jsou tedy k dispozici všechny informace. Na této ploše nelze popsat to, co je na stovkách stran projektů, které jsem měl možnost oponovat. Jedná se o výsledky dvouletého období kolegy Klímy, kdy se uzavřel na své chatě a tyto nové koncepty vynalezl (bočním výsledkem byl návrh nejrychlejší metody hledání kolizí MD5). Domníváme se, že hlavní přínosy jsou:

**Zodpovězení otázky, proč mají současné hašovací funkce problémy.** Tento problém Klíma definoval jako první na světě a z důvodu zaneprázdnění prezentoval poněkud skromně pouze na MKB (link viz níže) v češtině v prosinci minulého roku [4].

**Návrh stavby nové generace hašovacích funkcí.** Kolega Klíma navrhl něco neobvyklého – blokovou šifru, jejíž klíč může protivník znát. Podobná myšlenka v roce 1975 v jiné souvislosti znamenala revoluci v kryptografii a založila nový obor – kryptografii s veřejným klíčem. Speciální blokové šifry mají mnohem tvrdší požadavky – útočník může šifrovací klíč sám volit a libovolně s ním manipulovat. Po tomto úvodu je zřejmé, že se jedná o silnější šálek kávy a je snad jasnější, proč se NBÚ rozhodl tento koncept povolit publikovat.

**Návrh třídy speciálních blokových šifer.** V době zveřejnění první práce – nové stavby hašovacích funkcí [5] – si mnozí kryptologové nemohli představit žádný praktický příklad speciální blokové šifry, neboť v té době byl ještě neveřejný, a publikovaná koncepce mohla působit jako neužitečná teorie. Dnes je popis speciálních blokových šifer již k dispozici, dokonce v [1] je popis skládačky, ze které lze speciální blokové šifry stavět. Pozoruhodné je, že tato skládačka umožňuje si namíchat svoji šifru. Do vzorce, který reprezentuje bezpečnost takové šifry, pak stačí jen dosadit konkrétní hodnoty zvolených prvků. Konkrétní navrhované hodnoty u funkcí DN a HDN (10 rund) jsou navrženy s velkou bezpečnostní rezervou (byly navrhovány pro NBÚ), o které se může současným hašovacím funkcím jenom zdát.

**Možnost použít speciální blokovou šifru k šifrování.** Trochu podivné, když klíč může útočník znát. Tento koncept opravdu předbíhá dobu. Je to ale velmi jednoduché – pokud speciální bloková šifra odolává různým útokům i ze strany klíče a přidáme-li zpětně předpoklad, že tento klíč útočník nezná, dostaneme klasickou blokovou šifru s přídavnými bezpečnostními opatřeními. I u klasických blokových šifer se totiž začínáme setkávat s útoky, které odhalují některé bity klíče nebo je nedokáží odhalit, ale umí je měnit (to vše umí dnes zcela reálně postranní kanály). Čili u klasické blokové šifry se dnes do jisté míry narušuje jak předpoklad neznalosti klíče útočníkem, tak předpoklad, že není schopen s ním manipulovat. To jsou věci, které dříve byly nemyslitelné. Speciální bloková šifra je velmi těžkým kalibrem proti těmto typům útoků. Z jejího původního poslání – být stavebním blokem hašovací funkce – se může vrátit k poslání staronovému, a to šifrovat data.

Bude docela zajímavé sledovat, jak budou tyto myšlenky přijaty. Vědecký svět má dost času a umí být i krutý, takže je možné, že tato myšlenka zapadne a bude oprášena třeba po deseti letech. V každém případě od kryptografů vyžaduje přehodnocení jejich přístupu k hašovacím funkcím a oprostění se od starých schémat a vzorů.

Následující článek kolegy Klímy se bude věnovat už jen speciální blokové šifře DN a jejímu konkrétnímu použití v hašovací funkci HDN [1]. K dispozici jsou i zdrojové kódy a testovací příklady [2] a [3]. Pokud se budete chtít vrátit k teoretickému odůvodnění stavby nové generace hašovacích funkcí na bázi speciální blokové šifry, je to popsáno v [4] a [5].

## Literatura:

[1] Vlastimil Klíma: Rodina speciálních blokových šifer DN a hašovacích funkcí nové generace HDN typu SNMAC, IACR ePrint archive Report 2007/050, February, 2007

[2] Testy funkcí DN a HDN v jazyce C, podle příspěvku naprogramoval Milan Zámotny. Freeware.

[3] Zdrojový kód speciální blokové šifry DN a hašovací funkce HDN, vyjmutý z příspěvku, neoptimalizovaný, včetně testů rychlosti.

[4] Vlastimil Klíma: Hašovací funkce nové generace SNMAC, Mikulášská kryptobesídka MKB 2006, Praha, Hotel Olympik, 7. – 8. prosinec 2006, prezentace a text příspěvku.

[5] Vlastimil Klíma: Nový koncept hašovacích funkcí SNMAC s využitím speciální blokové šifry a konstrukcí NMAC/HMAC, IACR ePrint archive Report 2006/376, October, 2006

Tyto a další související práce (v češtině) a zdrojové kódy šifer a hašovacích funkcí jsou na domácí stránce projektu [http://cryptography.hyperlink.cz/SNMAC/SNMAC\\_CZ.html](http://cryptography.hyperlink.cz/SNMAC/SNMAC_CZ.html).

## B. Rodina speciálních blokových šifer DN a hašovacích funkcí nové generace HDN typu SNMAC

RNDr. Vlastimil Klíma, nezávislý konzultant,

v.klima@volny.cz, <http://cryptography.hyperlink.cz>

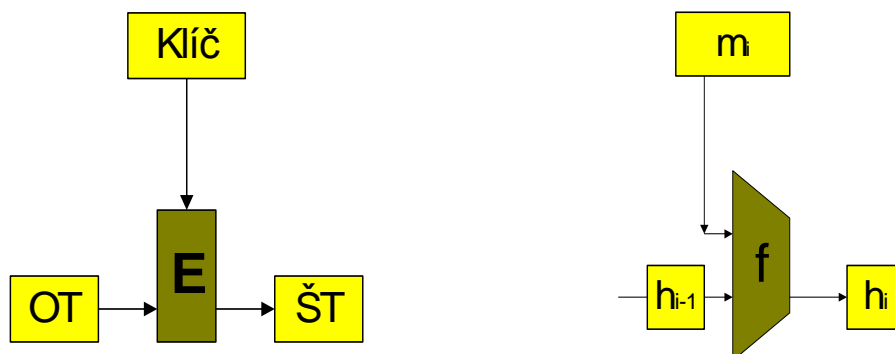
**Abstrakt.** Speciální bloková šifra je nové kryptografické primitivum, které bylo navrženo jako stavební prvek hašovacích funkcí nové generace SNMAC [KI06]. Na rozdíl od klasické blokové šifry předpokládá, že útočník zná šifrovací klíč a může s ním libovolně manipulovat. Hašovací funkce SNMAC mají veřejně známá návrhová kritéria, limitně se blíží náhodnému orákulu, jsou výpočetně odolné proti nalezení vzoru a kolize a umožňují návrh pomocí různých instancí speciálních blokových šifer.

V tomto příspěvku prezentujeme rodinu speciálních blokových šifer Double Net DN(n, k)- $\rho$  s n bitovým blokem, k bitovým klíčem a počtem rund  $\rho$ , principy tvorby jejich stavebních prvků a návrhová kritéria. Na bázi DN definujeme rodinu hašovacích funkcí HDN(n, k)- $\rho$  s n bitovým hašovacím kódem, která hašuje po blocích o délce k - n bitů.

Jako příklad uvádíme definice DN(512, 8192)-10 a HDN(512, 8192)-10. Jsou to prakticky použitelné funkce, jejichž rychlost je 2-3 krát nižší než rychlost SHA-512 a Whirlpool.

Speciální blokovou šifru můžeme použít klasicky k šifrování. Má výhodu, že bude připravena na nejrůznější útoky ze strany klíče, které se u klasických blokových šifer teprve rozvíjejí. Jsou to útoky postranními kanály, útoky příbuznými klíči, pravoúhelníkové útoky a jiné (viz například [Bi93], [Bi03], [Ki04], [Ho05], [Ki05], [Bi05], [Bi06]). Tyto útoky budou vznikat stále častěji s rozšiřováním kryptografických metod a prostředků. Všechny mají společné to, že původní předpoklad o neznalosti klíče protivníkem nebo o nemožnosti s ním manipulovat oslabují nejrozmanitějšími způsoby. Obranu proti nim dokládá i vývoj funkcí, které zpracovávají klíč, od funkcí typu COPY u DES a TripleDES ke slabě nelineárním funkcím u AES. Použití speciálních blokových šifer pro šifrování dat není dnes ještě vidět jako nezbytné, ale v budoucnu pravděpodobně bude. U hašovacích funkcí je to nezbytné už dnes.

Domníváme se, že příčinou současných problémů hašovacích funkcí je to, že jako kompresní funkci používají klasickou blokovou šifru, která byla původně navrhována ke zcela jiným účelům. Hlavní rozpory ukazuje následující obrázek a tabulka [KI06a].



<i>klasická bloková šifra E</i>	<i>kompresní funkce f</i>
<i>obsahuje prvek neznámý útočnickovi</i>	útočník zná všechny vstupy kompresní funkce, může s nimi manipulovat
<i>je určena k zakrytí struktury a obsahu otevřeného textu v šifrovaném textu na základě tajného prvku, neznámého útočnickovi (tedy ve výstupu zakrývá strukturu a obsah části vstupu na základě neznalosti jiné části vstupu)</i>	je určena k zakrytí struktury a obsahu celého vstupu ve výstupu, je založena na veřejné funkci
<i>při fixovaném klíči je permutací</i>	je to náhodné zobrazení
<i>je invertibilní</i>	požadavek neinvertibility (jednocestnosti) je zcela zásadní
<i>je snadné vytvářet kolize</i>	požadavek bezkoliznosti je zcela zásadní

Proto vznikla speciální bloková šifra.

## 1. Úvod

U hašovacích funkcí využívajících blokové šifry v kompresní funkci má útočník možnost manipulace s otevřeným textem i klíčem. Klasické blokové šifry však nejsou cíleně konstruovány tak, aby těmto útokům primárně odolávaly – jistá odolnost zde je, avšak lze ji v podstatě označit za vedlejší efekt. Nová generace hašovacích funkcí SNMAC [KI06] proto využívá v kompresní funkci speciální blokovou šifru. Jakmile bude koncept speciální blokové šifry prozkoumán a přijat v hašovacích funkcích, není důvodu, proč ho v předstihu nepoužít také jako primitivum pro původní účel šifrování dat. Navrhujeme v budoucnu přejít od klasických blokových šifer k více bezpečným a univerzálním speciálním blokovým šifrům.

Klasická bloková šifra je kryptografickým primitivem, které má chránit obsah a strukturu otevřeného textu v šifrovaném textu, a to s využitím utajeného šifrovacího klíče.

Ve stavbě klasické blokové šifry se neznalosti klíče útočníkem využívá zásadním způsobem k dosažení vysoké rychlosti šifrování. Klíč se prakticky nijak neupravuje. Tzv. fáze přípravy klíče (expanze klíče) je u většiny klasických blokových šifer velmi jednoduchá. Například u DES je využita pouze funkce "kopíruj". U AES je použita slabě nelineární transformace. U většiny blokových šifer jsou použity slabě nelineární nebo jednoduché funkce.

U současných hašovacích funkcí se v kompresní funkci používá klasická bloková šifra, její myšlenky a konstrukce. Při útocích na hašovací funkce tříd MD a SHA bylo kromě jiného využito zásadně faktu, že pro zpracování klíče jsou použity slabě nelineární funkce. Ty umožnily na mnoha místech přesně modifikovat vnitřní stav hašovací funkce podle předem zadaného plánu (diferenční cesty). Silně nelineární funkce by toto neumožnily.

U klasických blokových šifer se nejprve předpokládalo, že útočník nezná otevřený text, později se připustilo, že může znát jeho části, později, že může některé části otevřeného textu volit. Nyní se předpokládá jakákoliv manipulovatelnost s otevřeným textem i šifrovaným textem. Proti těmto možnostem útočníka se konstruovaly silně nelineární funkce, zpracovávající otevřený text.

Avšak také se předpokládalo a stále předpokládá, že útočník nezná šifrovací klíč a nemá žádnou možnost s ním manipulovat. Rozvojem technologií a vznikem různých forem šifrovacích zařízení (čipové karty, servery SSL, kryptografické moduly, knihovny,...) vznikly

útočníkům nové možnosti, které oslabují oba dva původní předpoklady – neznalost klíče i nemožnost s ním manipulovat.

Tyto možnosti ukázaly zejména postranní kanály nejrůznějších typů (chybové, napěťově-proudové, elektromagnetické,...), kdy je možná jak manipulace s klíčem, tak jeho částečná znalost.

Klíč je dnes zpracováván lineárně nebo slabě nelineárně a není proti podobným útokům chráněn. Vývoj v dalších desítkách let nepochybně ukáže podobný posun i v útocích ze strany klíče. Máme-li konstruovat kvalitní blokové šifry do budoucna, bude vhodné zesílit funkce, které jsou použity ke zpracování klíče a volit je stejně kvalitní funkce a stejně odolné proti diferenciální a lineární kryptoanalýze a dalším útokům, jako funkce pro zpracování otevřeného textu. Než se útoky ze strany klíče plně projeví, může trvat desítky let. Je proto otázkou, kdy k těmto obranným opatřením přistoupit.

Možnosti manipulace s klíčem vplynuly plně na povrch, když se klasická bloková šifra použila v hašovacích funkcích. Protože u kompresní funkce neexistuje žádný utajený prvek, útočník má možnost manipulovat se všemi vstupy použité blokové šifry, tedy i s jejím klíčem. U hašovacích funkcí musíme k těmto opatřením přistoupit neprodleně, neboť tyto možnosti má útočník už dnes.

Z tohoto důvodu byla navržena speciální bloková šifra pro hašovací funkce a koncepce hašovacích funkcí nové generace SNMAC. V tomto příspěvku popisujeme první třídu speciálních blokových šifer DN a na nich založených hašovacích funkcí HDN. Plná verze příspěvku je uvedena v [KI07].

## 2. Popis rodiny funkcí Double Net

V této kapitole uvedeme popis rodiny blokových šifer Double Net  $DN(n, k)-\rho$ , principy tvorby jejich stavebních prvků a návrhová kritéria.

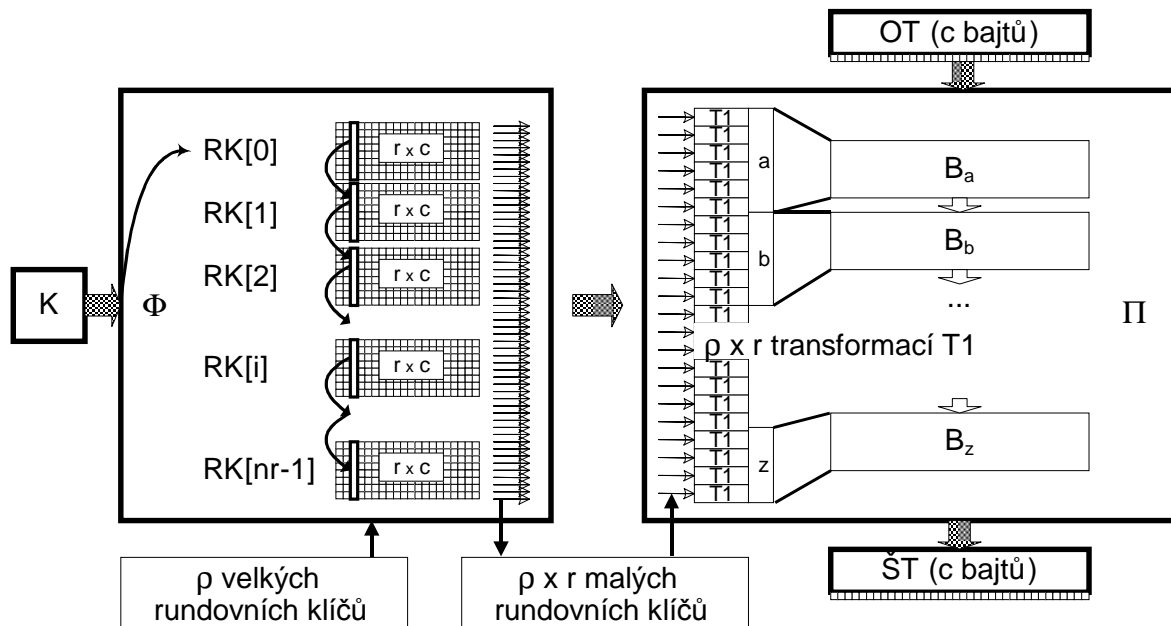
### 2.1 Základní schéma $DN(n, k)-\rho$

$DN(n, k)-\rho$ , je bloková šifra, která má blok o délce  $n$  bitů, šifrovací klíč  $K$  o délce  $k$  bitů a  $\rho$  velkých rund, kde  $\rho^{**}$  je bezpečnostní parametr.

DN se skládá ze dvou funkcí, expanze klíče  $\Phi$  a součinové šifry  $\Pi$ . Základní myšlenkou dvojité sítě DN je to, že klíče  $a, b, \dots$ , pro dílčí šifry součinové šifry  $\Pi = B_z \bullet \dots \bullet B_b \bullet B_a$  jsou samy vytvářeny kvalitní blokovou šifrou  $\Phi$ . Se zvyšováním počtu rund se klíče  $(a, b, \dots)$  a  $(\dots, y, z)$  stávají výpočetně neodlišitelnými od nezávislých (náhodných veličin) neboť odpovídají vztahu otevřeného a šifrovaného textu blokové šifry  $\Phi$ . Potom i blokové šifry  $(B_a, B_b, \dots)$  a  $(\dots, B_y, B_z)$  se stávají výpočetně neodlišitelnými od nezávislých (náhodných) blokových šifer. Efektivita je přitom dosaženo tím, že funkce  $\Phi$  je kvalitní blokovou šifrou pouze ve sloupcových řezech pole RK. Promíchání sloupců pole RK mezi sebou a s otevřeným textem zajistí funkce  $\Pi$ . Tento proces je tím efektivnější, čím více je sloupců v poli rundovních klíčů.

---

\*\* Proměnná  $\rho$  je v programovém kódu (a v obrázcích) označována jako  $\rho$ .



Obr. 1: Rodina funkcí DN

Délka bloku a délka klíče jsou zarovnány na bajty, délka klíče je násobkem délky bloku a délka bloku je násobkem 32 bitů. Schéma je popsáno na úrovni bajtů. Počet bajtů otevřeného textu označujeme  $c = n/8$ . Je to také počet sloupců v poli klíčů. Počet bajtů klíče  $K$  je  $k/8$ . Bajty klíče jsou vepsány do pole o rozměru  $r$  řádků a  $c$  sloupců zleva doprava a shora dolů, kde  $r = k/n$  ( $r \times c = k/n \times n/8 = k/8$ ). Funkce  $\Phi$  expanduje šifrovací klíč na pole rundovních klíčů. Pracuje v trojrozměrném poli  $\rho \times r \times c$  bajtů  $RK[i][j][t]$ ,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $t = 0, \dots, c - 1$ , které nazýváme polem rundovních klíčů. První index ( $i$ ) určuje velký rundovní klíč  $RK[i]$  jako dvourozměrné pole o rozměru  $r \times c$ . Velký rundovní klíč  $RK[i]$  se skládá z  $r$  malých rundovních klíčů  $RK[i][j]$ ,  $j = 0, \dots, r - 1$ . Malý rundovní klíč  $RK[i][j]$  je jeden řádek velkého rundovního klíče a má  $c$  bajtů  $RK[i][j][t]$ ,  $t = 0, \dots, c - 1$ . Vstupem funkce  $\Phi$  je klíč  $K$ , který je vepsán do prvního velkého rundovního klíče  $RK[0]$  (zleva doprava a shora dolů). Funkce  $\Phi$  vytváří z prvního velkého rundovního klíče  $RK[0]$  postupně dalších  $\rho - 1$  velkých rundovních klíčů  $RK[i]$ ,  $i = 1, \dots, \rho - 1$ .

Funkce  $\Pi$  míchá otevřený text s polem rundovních klíčů, viz obr. 1. Primárně je  $\Pi$  součinem  $\rho \times r$  elementárních transformací  $T1$ ,  $\Pi = \Pi_{i=\rho-1, \dots, 0} \Pi_{j=r-1, \dots, 0} T1_{i,j}$ , kde každá transformace  $T1_{i,j}$  používá jeden malý rundovní klíč  $RK[i][j]$ ,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ . Pokud sdružíme několik těchto transformací  $T1$  (například  $r/2$ ,  $r$  nebo  $2r$ ) do jedné blokové šifry  $B$ , můžeme funkci  $\Pi$  chápat jako součin blokových šifer  $B$ , z nichž každá využívá několik malých rundovních klíčů, tj.  $\Pi = B_z \cdot \dots \cdot B_b \cdot B_a$ , kde  $z \parallel \dots \parallel b \parallel a = RK = RK[\rho - 1][r - 1] \parallel RK[\rho - 1][r - 2] \parallel \dots \parallel RK[0][1] \parallel RK[0][0]$ .

Transformace  $T1$  se skládá ze substituce a permutace na úrovni bajtů, z lineární transformace na úrovni bitů (lineární transformace nesmí být převeditelná na úroveň bajtů) a přičtení malého rundovního klíče a rundovní konstanty.

Z hlediska prokazování vlastností chápeme funkci  $\Pi$  jako součin blokových šifer  $B$ , z hlediska realizace v HW i SW jako  $\rho \times r$  malých rund  $T1$ .

## 2.2 Funkce $\Phi$

Vstupem funkce  $\Phi$  je šifrovací klíč  $K$  a výstupem je pole rundovních klíčů  $RK$ . Funkce  $\Phi$  se skládá ze **sloupcové transformace** a **závěrečné klíčové permutace**. Sloupcová transformace naplňuje pole  $RK$  a závěrečná klíčová permutace provádí permutaci bajtů v tomto poli. Sloupcová transformace je systém  $c$  nezávislých sloupcových transformací  $F_t$ ,  $t = 0, \dots, c - 1$ , které pracují ve sloupcích pole  $RK$ . Každá sloupcová transformace je součinovou blokovou šifrou  $F_t = f_{\rho-1,t} \bullet \dots \bullet f_{2,t} \bullet f_{1,t}$  s délkou bloku  $r$  bajtů, přičemž její jednotlivé rundy nazýváme dílčí sloupcové transformace ( $f_{i,t}$ ). Sloupec  $t$  pole  $RK$  se tak postupně naplňuje výsledky dílčích rund blokové šifry  $F_t$ . Každá z  $(\rho - 1) \times c$  dílčích sloupcových transformací  $f_{i,t}$ ,  $i = 1, \dots, \rho - 1$ ,  $t = 0, \dots, c - 1$  je elementární transformací (T2), která se skládá ze substituce na úrovni bajtů ( $r$  substitučních boxů SubsF), lineární transformace na úrovni bitů (pomocí matice typu MDS o rozměru  $r \times r$ ) a přičtení  $r$ -bajtové rundovní konstanty (RConstF). Každá sloupcová transformace  $F_t$  je tak ve skutečnosti blokovou šifrou s konstantním klíčem (rundovní klíče jsou konstanty), viz obr. 2.

*Zápis šifrovacího klíče do pole  $RK$ :*

Klíč  $K$  se zapíše do pole bajtů  $RK[0]$  o rozměru  $r \times c$  zleva doprava a shora dolů:

$$RK[0][j][t] = K[j \cdot c + t], j = 0, \dots, r - 1, t = 0, \dots, c - 1.$$

*Vytvoření pole  $RK$ :*

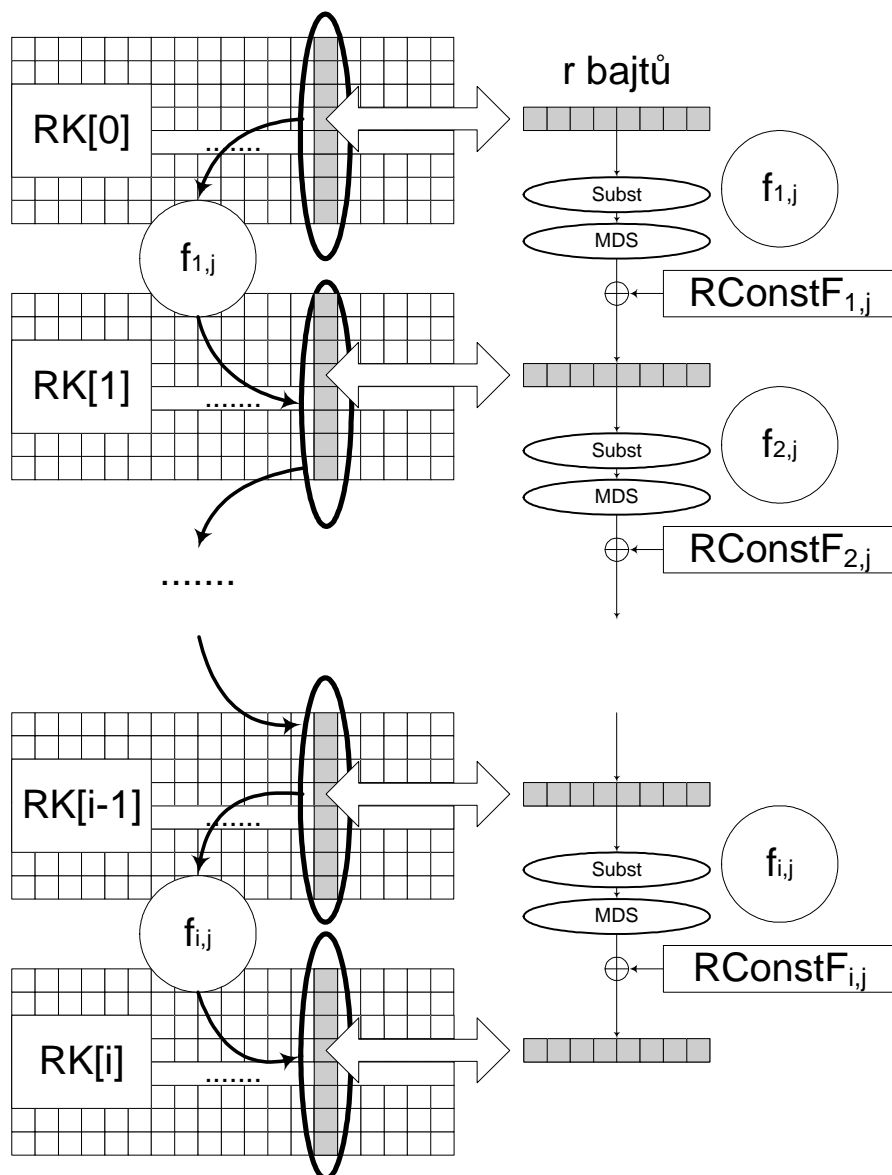
Bajt  $RK[i][j][t]$  označujeme krátce jako  $RK_{i,j,t}$ .

Rundovní klíče  $RK[0], \dots, RK[\rho - 1]$  se vytváří odděleně po sloupcích ( $t = 0, \dots, c - 1$ ) pomocí funkcí  $F_t = f_{\rho-1,t} \bullet \dots \bullet f_{2,t} \bullet f_{1,t}$  postupně takto:  $RK[0] \rightarrow RK[1] \rightarrow \dots \rightarrow RK[\rho - 1]$ . Každá funkce  $f_{i,t}$  používá  $r$  (obecně různých) substitučních boxů  $\text{SubsF}_{i,j,t}$ ,  $j = 0, \dots, r - 1$ , matici  $\text{MDS}_{i,t}$  typu  $r \times r$  a  $r$  bajtovou rundovní konstantu  $\text{RConstF}_{i,t} = (\text{RConstF}_{i,0,t}, \text{RConstF}_{i,1,t}, \dots, \text{RConstF}_{i,r-1,t})$ . Pro  $i = 1, \dots, \rho - 1$  a  $t = 0, \dots, c - 1$  máme  $(RK_{i,0,t}, RK_{i,1,t}, \dots, RK_{i,r-1,t}) = f_{i,t}(RK_{i-1,0,t}, RK_{i-1,1,t}, \dots, RK_{i-1,r-1,t}) = (\text{MDS}_{i,t} \bullet (\text{SubsF}_{i,0,t}(RK_{i-1,0,t}), \text{SubsF}_{i,1,t}(RK_{i-1,1,t}), \dots, \text{SubsF}_{i,r-1,t}(RK_{i-1,r-1,t}))^T)^T \oplus (\text{RConstF}_{i,0,t}, \text{RConstF}_{i,1,t}, \dots, \text{RConstF}_{i,r-1,t})$ , kde operátor  $^T$  znamená transpozici řádku na sloupec a naopak. Matice  $\text{MDS}_{i,t}$  je maticí typu MDS (maximum distance separable) a násobení je prováděno v tělese  $\text{GF}(2^8)$ .

*Závěrečná klíčová permutace  $\text{KeyPerm}$ :*

Závěrečná klíčová permutace je permutací na množině  $\text{INDX} = \{0, 1, \dots, \rho - 1\} \times \{0, 1, \dots, r - 1\} \times \{0, 1, \dots, c - 1\}$ ,  $\text{KeyPerm}: \text{INDX} \rightarrow \text{INDX}: (i, j, t) \rightarrow \text{KeyPerm}(i, j, t)$ . Permutuje bajty v poli  $RK$ , tj.  $RK_{i,j,t} = RK_{\text{KeyPerm}(i,j,t)}$ ,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $t = 0, \dots, c - 1$ . Aplikuje se po vytvoření celého pole  $RK$  sloupcovou transformací. Tato permutace není z bezpečnostního hlediska povinná, jejím cílem je zefektivnit difúzi sloupců rundovních klíčů uvnitř funkce  $\Pi$ . Permutace může být velmi jednoduchá, například cyklický posun bajtů v rámci malého rundovního klíče. Podrobnosti o konstrukci jsou uvedeny dále.





Obr.2: Sloupcová transformace

## 2.3 Funkce $\Pi$

Funkce  $\Pi$  je blokovou šifrou. Otevřený text tvoří  $c$  bajtů:  $indata(0), \dots, indata(c - 1)$ . Šifrový text tvoří  $c$  bajtů:  $outdata[0], \dots, outdata(c - 1)$ . Šifrovacím klíčem je pole  $RK$ , obsahující  $\rho \times r$  malých rundovních klíčů  $RK[i][j]$ ,  $i = 0, 1, \dots, \rho - 1, j = 0, 1, \dots, r - 1$ . Primárně je  $\Pi$  součinem  $\rho \times r$  elementárních transformací  $T1$ ,  $\Pi = \Pi_{i=\rho-1, \dots, 0} \Pi_{j=r-1, \dots, 0} T1_{i,j}$ , kde  $T1_{i,j}$ , používá malý rundovní klíč  $RK[i][j]$ ,  $i = 0, \dots, \rho - 1, j = 0, \dots, r - 1$ . Výstup z jedné transformace  $T1$  je vstupem do další transformace  $T1$ . Vstup do funkce  $\Pi$  je vstupem do první transformace  $T1$ , výstup z poslední transformace  $T1$  je výstupem z funkce  $\Pi$ .

### 2.3.1 Transformace $T1_{i,j}$

Každá transformace  $T1_{i,j}$ ,  $i = 0, \dots, \rho - 1, j = 0, \dots, r - 1$ , se skládá ze substituce a permutace na úrovni bajtů, z lineární transformace na úrovni bitů a (binárního) přičtení malého rundovního

klíče a rundovní konstanty. Všechny tyto proměnné mohou být pro různé transformace  $T_{i,j}$  různé. Pro každou dvojici  $(i, j)$ ,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ , máme:

- $c$  substitučních boxů  $\text{SubsB}_{i,j,t}$ ,  $t = 0, \dots, c - 1$ , převádějících bajt na bajt
- permutaci na množině  $\{0, 1, \dots, c - 1\}$ , kterou nazýváme permutací typu "Small-Middle-Large" a označujeme  $\text{SMLPerm}_{i,j}: \{0, 1, \dots, c - 1\} \rightarrow \{0, 1, \dots, c - 1\}: t \rightarrow \text{SMLPerm}_{i,j}(t)$ ,
- lineární transformaci, která je tvořena  $n/32 = c/4$  maticemi  $\text{MDS}_{i,j,v}$  typu MDS (maximum distance separable code) o rozměru  $4 \times 4$  bajty,  $v = 0, \dots, c/4 - 1$ ,
- rundovní konstantu  $\text{RConstB}_{i,j}$  o  $c$  bajtech ( $\text{RConstB}_{i,j,0}, \dots, \text{RConstB}_{i,j,c-1}$ ),
- malý rundovní klíč  $\text{RK}[i][j]$  o  $c$  bajtech ( $\text{RK}_{i,j,0}, \dots, \text{RK}_{i,j,c-1}$ ).

**Poznámka k lineární transformaci v T1.** Lineární transformace může být obecnější, v konstrukci DN se využívá možnost realizace lineární úrovně (malými) maticemi typu  $4 \times 4$ . Násobení maticí je prováděno v tělese  $\text{GF}(2^8)$ . Z použití těchto matic vyplývá požadavek, aby otevřený text byl násobkem 32 bitů. Pokud jako stavební prvek použijeme jiné lineární matice, nemusí být otevřený text násobkem 32 bitů.

Tím je popis DN ukončen.

## 2.4 Volitelné parametry třídy blokových šifer DN

Schéma DN je obecným schématem, založeným na použití dvou SP sítí  $\Phi$  a  $\Pi$ . Jedna SP síť expanduje šifrovací klíč na pole rundovních klíčů a druhá síť promíchává rundovní klíče s otevřeným textem. Oproti klasickým blokovým šifrám je klíč zpracován stejně kvalitně jako otevřený text.

Parametry DN jsou její stavební prvky, jejich typ, rozměr a obsah.

$\text{DN}(n, k) - \rho$  má volitelné tyto parametry:

Základní rozměry:

- $n$ , délka bloku otevřeného textu v bitech ( $c = n/8$ ),
- $k$ , délka šifrovacího klíče  $K$  v bitech ( $r = k/n$ ),
- $\rho$ , počet velkých rund,

Funkce  $\Phi$ :

- S-boxy  $\text{SubsF}_{i,j,t}$  převádějící bajt na bajt,  $i = 1, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $t = 0, \dots, c - 1$ ,
- matice  $\text{MDS}_{i,t}$  o rozměru  $r \times r$ ,  $i = 1, \dots, \rho - 1$ ,  $t = 0, \dots, c - 1$ ,
- konstanty  $\text{RConstF}_{i,t}$  o  $r$  bajtech,  $i = 1, \dots, \rho - 1$ ,  $t = 0, \dots, c - 1$ ,
- závěrečná klíčová permutace **KeyPerm** na množině  $\{0, \dots, \rho - 1\} \times \{0, \dots, r - 1\} \times \{0, \dots, c - 1\}$ ,

Funkce  $\Pi$ :

- permutace  $\text{SMLPerm}_{i,j}$  na množině  $\{0, \dots, c - 1\}$ ,
- S-boxy  $\text{SubsB}_{i,j,t}$  převádějící bajt na bajt,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $t = 0, \dots, c - 1$ ,
- matice  $\text{MDS}_{i,j,v}$  o rozměru  $w \times w$ ,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $v = 0, \dots, c/w - 1$ , kde  $w$  je nějaký dělitel čísla  $c$  (každá z matic může mít jiný rozměr, zejména se bude využívat  $w = 4$ , podrobněji viz následující kapitola),
- konstanty  $\text{RConstB}_{i,j}$  o  $c$  bajtech,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ .

**Poznámka.** Uvedené parametry a stavební prvky mohou být voleny s velkou volností. Pravidla, která musí tyto stavební prvky splňovat, lze předběžně a stručně shrnout takto:

- funkce  $\Pi$  je kvalitní bloková šifra,
- všechny sloupcové transformace funkce  $\Phi$  jsou kvalitní blokové šifry (s konstantním klíčem), pokud možno odlišné

- funkce  $\Phi$  a  $\Pi$  používají odlišné S-boxy,
- všechny S-boxy mají dobré lineární a diferenciální charakteristiky a jsou generovány nealgebraicky, nejlépe (pseudo)náhodně,
- matice ve funkci  $\Phi$  a  $\Pi$  jsou všechny typu MDS (maximum distance separable).

Podrobně jsou pravidla definována v následující kapitole.

### 3. Konstrukce sítě $\Pi$

#### 3.1 $\Pi$ jako součin blokových šifer B

Funkci  $\Pi$  konstruujeme jako součin blokových šifer B, z nichž každá využívá několik rund T1 (několik malých rundovních klíčů), tj.  $\Pi = B_z \bullet B_y \bullet \dots \bullet B_b \bullet B_a$ . Cílem je, aby  $\Pi$  byla kvalitní bloková šifra, odolná proti lineární a diferenciální kryptoanalýze. Blokové šifry  $B_z, \dots, B_a$  je možné konstruovat stejné, eventuálně je možno konstruovat stejné  $B_y = \dots = B_a (= B)$  a  $B_z$  může obsahovat "zbytkový počet malých rund". Blokovou šifru B s délkou bloku c bajtů konstruujeme primárně tak, aby byla co nejvíce odolná proti lineární a diferenciální kryptoanalýze. K tomu využijeme důkazů odolnosti SP sítí proti lineární a diferenciální kryptoanalýze z Dodatku A. Funkci B konstruujeme jako několikanásobně vnořenou SP síť. Vnořenými sítěmi se zabývaly práce ([Ho00], [Ka01], [Chu03], [Sa03]), ale zde postačí použít výsledky z [Ho00]. Z Vět 1 a 2 obdržíme odhady pravděpodobností maximálního diferenciálu ( $DP^B$ ) a lineárního obalu ( $LP^B$ ) blokové šifry B. Bloková šifra B je jednou rundou součinné šifry  $\Pi$ , takže odhad  $DP^B$  a  $LP^B$  vypovídá o kvalitě funkce  $\Pi = B_z \bullet B_y \bullet \dots \bullet B_b \bullet B_a$ . K odhadu  $DP^\Pi$  a  $LP^\Pi$  nelze přímo použít součin  $DP^B \times DP^B \times \dots \times DP^B \times DP^B$  ani  $LP^B \times LP^B \times \dots \times LP^B \times LP^B$ , i když dříve se to tak dělalo, ale postačuje, pokud  $DP^B$  a  $LP^B$  budou malé. Poznamenejme, že podle [NyKn92] lze k odhadu  $DP^\Pi$  pro  $\Pi = B \bullet B \bullet B \bullet B$  použít  $DP^B \times DP^B$ . Hodnota  $DP^\Pi$  je pravděpodobně nižší než uvedený (nejlepší současný) odhad  $DP^B \times DP^B$ , ale zatím chybí důkazové metody. Lze však očekávat, že tyto odhady se zpřesní a zlepší.

#### 3.2 S-boxy v síti $\Pi$

Poznamenejme, že všechny S-boxy v síti  $\Pi$  mohou být různé. Označme  $p_B$  ( $q_B$ ) maximum z hodnot maximální diferenciální pravděpodobnosti p (resp. maximální lineární pravděpodobnosti q) přes všechny S-boxy, použité ve funkci B. Čím menší jsou hodnoty  $p_B$  a  $q_B$ , tím větší odolnost proti lineární a diferenciální kryptoanalýze funkce B má.

#### 3.3 Příklad sítě $\Pi$ pro $n = 512$

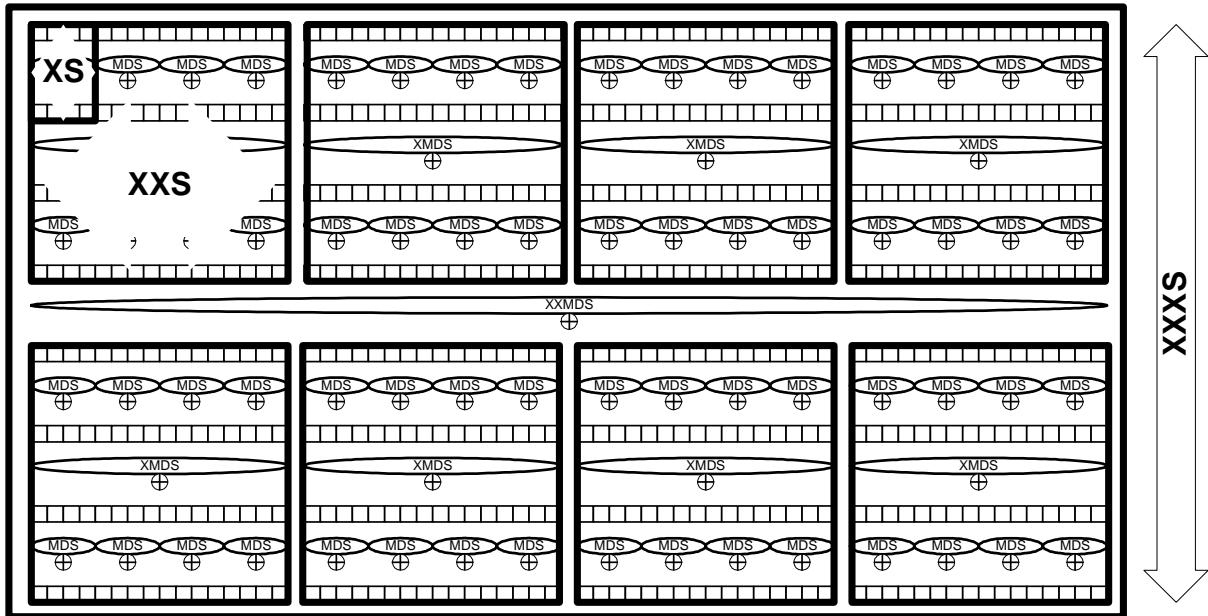
Šířka bloku  $n = 512$  bitů, tj.  $c = 64$  bajtů. Ke konstrukci  $\Pi$  použijeme rozklad  $c = 64 = c_1 \times c_2 \times c_3 = 4 \times 4 \times 4$ . Blokovou šifru B konstruujeme jako 3-úrovňovou vnořenou SPN síť.

XS-box konstruujeme jako SDS z S-boxů o šířce  $c_1 = 4$ ,

XXS-box konstruujeme jako SDS síť z XS-boxů o šířce  $c_2 = 4$ ,

XXXS-box konstruujeme jako SDS síť z XXS-boxů o šířce  $c_3 = 4$ .

XXXS-box je zároveň blokovou šifrou B. Skládá se z 8 základních transformací T1.



Obr.3: Blokovaná šifra B jako XXXS-box

Poznámka. Všechny S-boxy, všechny XS-boxy a všechny XXS-boxy mohou být různé.

Předpokládejme, že u všech boxů XS, XXS a XXXS máme zajištěnu maximalitu difúzní úrovně. Potom podle Věty 1 (viz Dodatek A), aplikované na SDS síť XS, XXS a XXXS, platí:

$$\begin{aligned} DP^{XS} &\leq (p_B)^4, \\ DP^{XXS} &\leq (DP^{XS})^4 \leq (p_B)^{4 \times 4}, \\ DP^{XXXS} &\leq (DP^{XXS})^4 \leq (p_B)^{4 \times 4 \times 4}, \end{aligned}$$

tedy

$$DP^B = DP^{XXXS} \leq (p_B)^{64} \text{ a analogicky podle Věty 2 (viz Dodatek A) dostáváme } LP^B = LP^{XXXS} \leq (q_B)^{64}, \text{ c.b.d.}$$

Tím je pro vhodná malá  $p_B$  a  $q_B$  zajištěna odolnost blokované šifry B proti DC a LC.

### 3.4 B jako N-úrovňová vnořená SPN

Konstrukce obecné sítě  $\Pi$  vychází z délky bloku otevřeného textu v bajtech  $c$ . Většinou bude  $c$  mocnina 2, zejména budou důležité hodnoty  $c = 8, 16, 32$  a  $64$ . Když konstruujeme B jako N-úrovňovou vnořenou SPN, vycházíme z rozkladu  $c = c_1 \times c_2 \times c_3 \times \dots \times c_N$ , kde  $c_1$  je šířka první sítě XS,  $c_2$  šířka druhé sítě XXS ( $X^2S$ ), ...,  $c_N$  je šířka poslední sítě XX...XS ( $X^N S$ ).

$X^1S$ -box konstruujeme jako SDS z S-boxů o šířce  $c_1$ ,

$X^2S$ -box konstruujeme jako SDS síť z XS-boxů o šířce  $c_2$ ,

... atd.

$X^N S$ -box konstruujeme jako SDS síť z  $X^{N-1}S$ -boxů o šířce  $c_N$ .

Pokud počet malých rund  $\Pi$  není dělitelný počtem rund blokované šifry B, zbytek rund označujeme jako část blokované šifry B ( $B_z$ ), tj.  $\Pi = B_z \bullet B \bullet \dots \bullet B \bullet B$ .

V konstrukci předpokládáme, že u všech boxů XS, ...,  $X^N S$  máme zajištěnu maximalitu jejich difúzní úrovně.

### 3.5 Odolnost sítě $\Pi$ proti DC a LC

Podle úvodu k této kapitole nemáme (z nedostatku důkazových metod) jinou možnost, než odolnost sítě  $\Pi$  proti DC a LC měřit čísly  $DP^B$  a  $LP^B$ .

Chápeme-li  $B$  jako velký box  $B: \{0, 1\}^n \rightarrow \{0, 1\}^n$ ,  $n = 8c$ , pak máme definovanu jeho maximální diferenciální a maximální lineární pravděpodobnost (viz Dodatek A) jako

$DP^B = \max DP^B(\Delta x \rightarrow \Delta y)$ , kde maximum se bere přes všechna  $\Delta x \neq 0$ ,  $\Delta x \in \{0, 1\}^n$ ,  $\Delta y \in \{0, 1\}^n$ ,

$LP^B = \max LP^B(\Gamma x \rightarrow \Gamma y)$ , kde maximum se bere přes všechna  $\Gamma x, \Gamma y \neq 0$ ,  $\Gamma x \in \{0, 1\}^n$ ,  $\Gamma y \in \{0, 1\}^n$ .

#### Věta 3. Odolnost blokové šifry $B$ proti DC a LC.

Konstruujeme-li  $B$  jako několikanásobně vnořenou SP síť (podle Dodatku A), pak platí

$$DP^B \leq (p_B)^c,$$

$$LP^B \leq (q_B)^c.$$

**Důkaz.** Vyplývá z induktivního použití Věty 1 na konstrukci boxů  $X^1S, \dots, X^NS$ . Máme  $DP^B = DP^{X^NS} \leq (DP^{X^{N-1}S})^{c_N} \leq (DP^{X^{N-2}S})^{c_{N-1} \times c_N} \leq \dots \leq (DP^S)^{c_1 \times \dots \times c_{N-1} \times c_N} = (DP^S)^c = (p_B)^c$ . Podobně podle Věty 2 dostáváme  $LP^B \leq (q_B)^c$ .

**Důsledek. Současný nejlepší odhad odolnosti  $\Pi$  proti DC.** Podle odstavce výše je současný nejlepší odhad  $DP^\Pi$  roven  $DP^B \times DP^B \leq (p_B)^c \times (p_B)^c = (p_B)^{2c}$ , pokud  $\Pi$  má alespoň čtyři bloky  $B$ , i když ve skutečnosti je odhad zcela určitě mnohem menší. Lze očekávat, že tyto odhady se zpřesní a zlepší.

**Důsledek. Současný nejlepší odhad odolnosti  $\Pi$  proti LC.** U odhad odolnosti  $\Pi$  proti LC můžeme vycházet pouze z toho, že máme odhad  $LP^B \leq (q_B)^c$  jako jedné "rundy" součinné šifry  $\Pi = B_z \bullet B \bullet \dots \bullet B \bullet B$ .

**Poznámka. Variantní konstrukce pro stejná  $c$ .** Konstrukce této sítě může mít několik variant i pro stejné hodnoty  $c$ . Závisí to na rozkladu čísla  $c$  i na možnosti realizovat difúzní úroveň v různých boxech různě.

**Poznámka. Počet rund  $B$ .** Počet rund blokové šifry  $B$  vyplývá z toho, že každá vyšší síť SDS zahrnuje dvě rundy, tvořené nižší sítí SDS. Proto počet rund (počet substitučních úrovní) je roven dvojnásobku počtu činitelů v rozkladu čísla  $c$ , tj.  $2N$ .

**Závěr.** S-boxy  $\text{Subs}B_{i,j,t}$  převádějící bajt na bajt ( $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$ ,  $t = 0, \dots, c - 1$ ) můžeme volit libovolně, různé nebo stejné. Ideální volba jsou náhodné nebo pseudonáhodné S-boxy, které mají dostatečnou odolnost proti lineární a diferenciální kryptoanalýze. Na velikosti čísel  $p_B$  ( $q_B$ ) závisí odolnost sítě  $\Pi$  proti DC a LC a volba počtu blokových šifer v součinu  $\Pi = B_z \bullet B \bullet \dots \bullet B \bullet B$ . S-boxy použité v síti  $\Pi$  by se měly odlišovat od S-boxů sítě  $\Phi$ . S-boxy by neměly mít algebraickou strukturu (například S-boxy AES mají algebraickou strukturu), i když není žádný přímý důkaz pro tuto vlastnost.

### 3.6 Maximalita difúzní úrovně sítě $\Pi$

Maximalitu difúzní úrovně v  $X^1S$ -boxech můžeme zajišťovat velkými maticemi MDS o rozměru  $C \times C$ , kde  $C = c_1 \times \dots \times c_{i-1} \times c_i$ . Například pro síť  $\Pi$  z příkladu máme  $c = 64 = c_1 \times c_2 \times c_3 = 4 \times 4 \times 4$  a matice  $X^3\text{MDS}$  by byla o rozměru  $64 \times 64$  bajtů. Realizace takových

matic je náročná na čas i paměť. Místo toho je maximalitu možné zajistit jinými způsoby. Třída funkcí DN nepředepisuje způsob zajištění maximality. Jeden ze způsobů nyní popíšeme.

Místo jedné MDS matice typu  $C \times C$ , kde  $C = c_1 \times \dots \times c_{i-1} \times c_i$  použijeme  $c_1 \times \dots \times c_{i-1}$  matic MDS typu  $c_i \times c_i$ . V případě, že  $c$  je mocninu dvojky, rozklad čísla  $c$  děláme tak, aby téměř všechny činitele byly rovny 4, až na eventuelně první činitel, který může být 2, 4 nebo 8. Použité matice tak mohou být typu  $2 \times 2$ ,  $4 \times 4$  a  $8 \times 8$ .

$X^{i-1}S$ -box obsahuje dvě vrstvy, z nichž každá obsahuje  $c_i$   $X^{i-1}S$ -boxů. Matice  $X^{i-1}MDS$  spojuje první vrstvu  $c_i$  boxů typu  $X^{i-1}S$  s druhou vrstvou  $c_i$  boxů typu  $X^{i-1}S$ . Každý  $X^{i-1}S$ -box má šířku  $c_1 \times \dots \times c_{i-1}$  bajtů.

Matici  $X^{i-1}MDS$  bychom tedy mohli konstruovat jako matici typu  $(c_1 \times \dots \times c_{i-1} \times c_i) \times (c_1 \times \dots \times c_{i-1} \times c_i)$ . Místo toho ji konstruujeme jako systém  $c_1 \times \dots \times c_{i-1}$  matic MDS typu  $c_i \times c_i$ . Každá z malých matic typu  $c_i \times c_i$  vybírá (libovolně) právě jeden bajt z každého z  $c_i$  vstupních  $X^{i-1}S$ -boxů. Na vstupu této matice je tak  $c_i$  bajtů. Ty jsou maticí transformovány na výstupních  $c_i$  bajtů, které jsou po jednom vedeny (na libovolné místo) do každého z  $c_i$  výstupních  $X^{i-1}S$ -boxů. Systém těchto matic vytváří maximální difúzní úroveň. (Předesíláme, že výběr pozic vstupních bajtů uvnitř vstupních  $X^{i-1}S$ -boxů do matic definuje příslušné permutace SMLPerm, viz dále.)

**Věta 4. Maximalita difúzní úrovně.** Matice  $X^{i-1}MDS$ , konstruovaná výše jako systém  $c_1 \times \dots \times c_{i-1}$  matic MDS typu  $c_i \times c_i$  je maximální difúzní úrovní v  $X^{i-1}S$ -boxu.

**Důkaz.** Předpokládejme změnu v  $k$   $X^{i-1}S$ -boxech na vstupu,  $1 \leq k \leq c_i$ . Poznamenejme, že změna v  $X^{i-1}S$ -boxu znamená, že dojde ke změně jednoho nebo několika jeho vstupních bajtů. Uvažujme první změněný bajt v prvním změněném  $X^{i-1}S$ -boxu na vstupu. Tento bajt je vstupem některé z  $c_1 \times \dots \times c_{i-1}$  matic MDS typu  $c_i \times c_i$ , dané difúzní úrovně. Označíme ji  $M$ . Označme  $s$  celkový počet změněných bajtů na vstupu matice  $M$ . Máme  $1 \leq s \leq k \leq c_i$ . Protože  $M$  je maticí typu MDS o rozměru  $c_i \times c_i$ , na jejím výstupu dojde ke změně nejméně v  $c_i + 1 - s$  bajtech. Máme  $c_i + 1 - s \geq c_i + 1 - k$ . Protože všechny bajty na výstupu matice  $M$  jdou do různých  $X^{i-1}S$ -boxů na výstupu difúzní úrovně, dojde ke změně alespoň  $c_i + 1 - k$   $X^{i-1}S$ -boxů na výstupu. Tím je maximalita difúzní úrovně  $X^{i-1}MDS$  dokázána.

**Závěr.** Matice  $MDS_{i,j,v}$  mohou mít různé rozměry ( $w \times w$ , kde  $w$  je nějaký dělitel čísla  $c$ ) a různý obsah. V síti  $\Pi$  může být použito na různých místech mnoho různých (nebo stejných) typů různých (nebo stejných) matic, a to i v jedné difúzní úrovni. Musí být pouze dodržena maximalita všech difúzních úrovní.

Dále, všechny použité matice MDS by měly zajistit difúzi na úrovni bitů (a nikoli bajtů jako celku), což například nesplňuje matice MDS, obsahující pouze prvky (hex.)  $0x00$  a (hex.)  $0x01$ . Prvků (hex.)  $0x00$  a (hex.)  $0x01$  by matice MDS měly obsahovat zcela minimální množství. V binárním vyjádření  $8r \times 8r$  by matice MDS neměla být ani příliš řídkou ani příliš pravidelnou. Měla by být co nejvíce náhodnou binární maticí o rozměru  $8r \times 8r$ . Ideální je, pokud všechny matice  $MDS_{i,j,v}$  jsou různé a vytvořeny náhodně. To je opatření proti algebraickým útokům. Není však striktně zakázáno použít všechny matice stejné.

### 3.7 Permutace typu Small-Middle-Large

V tomto odstavci popíšeme tvorbu permutací typu SMLPerm (Small-Middle-Large Permutation) a vysvětlíme pojem adjungované bajty.

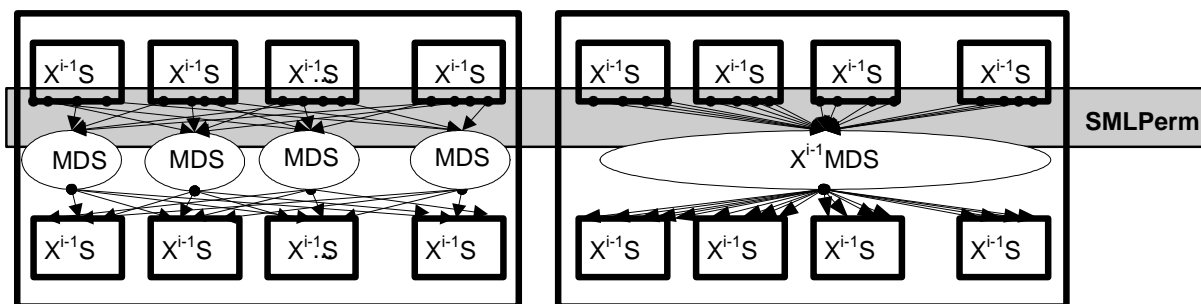
### 3.7.1 SMLPerm a T1

Pokud maximalitu difúzní úrovně  $X^{i-1}$ MDS (Small - mezi S-boxy, Middle - mezi XS boxy, Large - mezi  $X^{i-1}$ S-boxy) zajišťujeme největší možnou maticí typu  $(c_1 \times \dots \times c_{i-1} \times c_i) \times (c_1 \times \dots \times c_{i-1} \times c_i)$ , odpovídající permutace SMLPerm odpovídá pořadí výběru vstupních bajtů do této matice. V dané difúzní úrovni můžeme definovat jednu permutaci, v jiné difúzní úrovni můžeme definovat jinou permutaci. Permutaci můžeme také zahrnout přímo do matice. Máme tak možnost definovat jednu matici a různé permutace nebo různé matice (s permutovanými sloupci originální matice) a identické permutace.

Pokud maximalitu difúzní úrovně  $X^{i-1}$ MDS zajišťujeme pomocí  $c_1 \times \dots \times c_{i-1}$  (stejných) matic MDS typu  $c_i \times c_i$ , můžeme na jejich vstupy vést vstupy z  $c_i$   $X^{i-1}$ S-boxů také v různých permutovaných pořadích.

Maximalitu dané difúzní úrovně můžeme zajistit i maticemi jiných rozměrů.

Výběry bajtů do všech matic dané difúzní úrovně v celé šíři sítě  $\Pi$  definují permutaci  $c_1 \times \dots \times c_{N-1} \times c_N$  bajtů na  $c_1 \times \dots \times c_{N-1} \times c_N$  bajtů, kterou označujeme SMLPerm v této difúzní úrovni. Je to zároveň permutace v odpovídající transformaci T1. (Předesíláme, že výběr pozic u výstupních boxů je ve skutečnosti inverzí výběru vstupních pozic permutací SMLPerm v následující transformaci T1).



Obr.4: Permutace SMLPerm

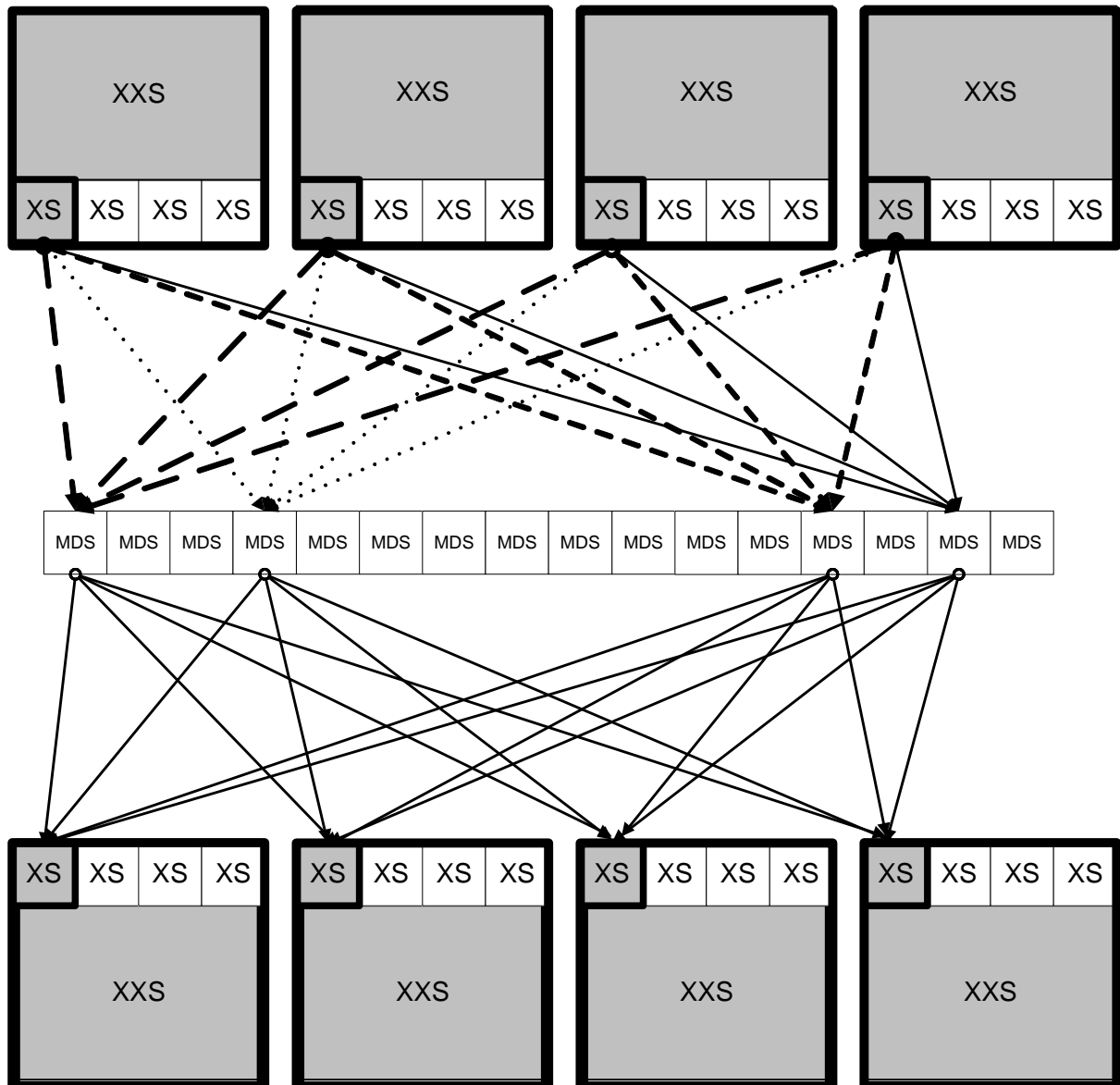
### 3.7.2 SMLPerm a různorodost

Pokud difúzní úroveň  $X^{i-1}$ MDS konstruujeme jako systém  $c_1 \times \dots \times c_{i-1}$  matic MDS typu  $c_i \times c_i$ , můžeme odpovídajícími permutacemi  $SMLPerm_{i,j}$  zlepšovat difúzi a zvyšovat různorodost (nesymetričnost) uvnitř blokové šifry B. Každá z malých matic MDS typu  $c_i \times c_i$  může libovolně vybírat právě jeden bajt z každého z  $c_i$  vstupních  $X^{i-1}S$ -boxů a výstup vést po jednom bajtu na libovolné místo do každého výstupního  $X^{i-1}S$ -boxu. To zajistí, že každý  $X^{i-1}S$ -box na vstupu ovlivní všechny  $X^{i-1}S$ -boxy na výstupu. O úroveň níže, u menších  $X^{i-2}S$ -boxů tomu tak být nemusí.

Každý z velkých  $X^{i-1}S$ -boxů na vstupu se skládá z  $c_{i-1}$  malých  $X^{i-2}S$ -boxů. Vezměme například první malý  $X^{i-2}S$ -box prvního velkého  $X^{i-1}S$ -boxu na vstupu a podívejme se na to, kolik ovlivňuje malých  $X^{i-2}S$ -boxů na výstupu. Tento box má  $c_1 \times \dots \times c_{i-2}$  bajtů, které pomocí  $c_i$  matic MDS ovlivňují  $c_1 \times \dots \times c_{i-2} \times c_i$  bajtů na výstupu. Z maximality vyplývá, že do každého z  $c_i$  výstupních  $X^{i-1}S$ -boxů vede právě  $c_1 \times \dots \times c_{i-2}$  výstupních bajtů. Tyto bajty mohou být rozmístěny v každém  $X^{i-1}S$ -boxu náhodně a zasahovat všechny jeho malé  $X^{i-2}S$ -boxy nebo vést v nejhorším případě pouze do jediného malého  $X^{i-2}S$ -boxu (má právě  $c_1 \times \dots \times c_{i-2}$  bajtů). Malé  $X^{i-2}S$ -boxy, které jsou ovlivněny na výstupu, nazýváme adjungované výstupní boxy (k danému malému boxu na vstupu). Ostatní vstupní bajty matic MDS, které zpracovávají daný malý box na vstupu, čerpají své vstupy také z několika dalších malých boxů na vstupu. Tyto boxy nazýváme adjungované vstupní boxy k danému boxu na vstupu. Podobně jako na výstupu může být i na vstupu k danému malému boxu adjungován v nejhorším případě pouze jeden malý  $X^{i-2}S$ -box z každého velkého  $X^{i-1}S$ -boxu. Tuto situaci

docílíme systematickou volbou, a sice, že do  $j$ -té matice MDS vedeme  $j$ -té bajty z každého velkého boxu ( $j = 0, \dots, c_1 \times \dots \times c_{i-2} \times c_{i-1} - 1$ ) a výstup vedeme na  $j$ -té pozici každého velkého výstupního boxu. Vzájemně jsou tak adjungované pouze vždy  $k$ -té malé boxy v rámci velkých boxů na vstupu  $i$  na výstupu ( $k = \lfloor j / (c_1 \times \dots \times c_{i-2}) \rfloor$ ,  $k = 0, \dots, c_{i-1} - 1$ ).

Systematický výběr permutací SMLPerm nemusí být proto pro difúzi nejlepší volbou. Ukážeme, že vhodnou volbou permutací lze dosáhnout rychlejší difúze a vyhnout se záměrným strukturálním pravidelnostem. Na následujících dvou obrázcích jsou dvě různé volby permutací. Obrázky ukazují, s jakými vstupními a výstupními malými boxy je adjungován první malý box v prvním velkém vstupním boxu.

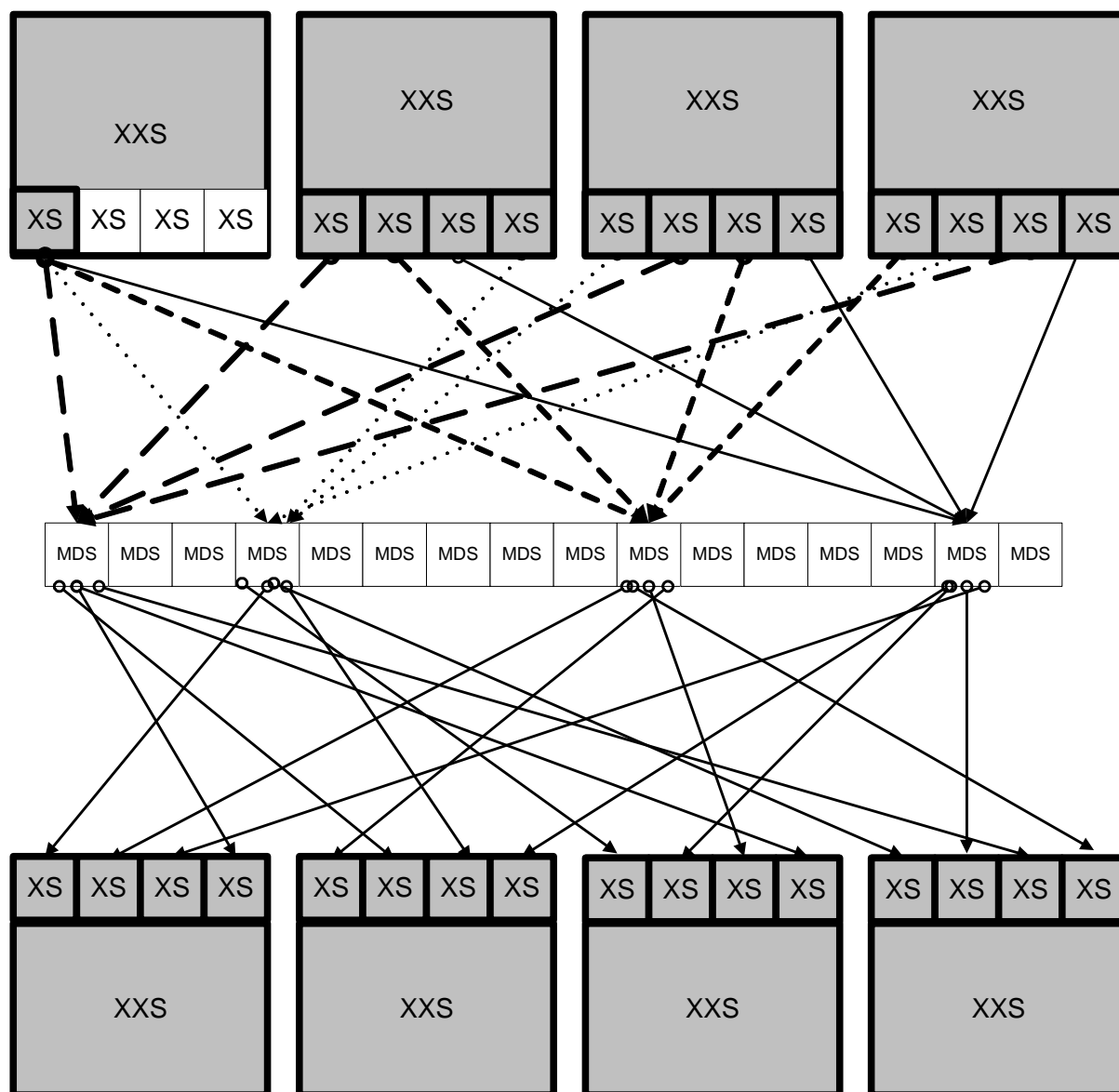


Obr.5: Systematická volba permutací

Na obrázku 5 jsou zvoleny permutace  $SMLPerm_{i,j}$  systematicky. První bajty prvních malých boxů jdou na vstup matice MDS a výstupy z ní jdou opět na první bajty prvních malých boxů v rámci velkých boxů. Podobně druhé, třetí a čtvrté bajty. Množina prvních malých boxů (všech velkých boxů) na vstupu tak prostřednictvím difúzní úrovně ovlivňuje pouze množinu prvních malých boxů (všech velkých boxů) na výstupu. Při této volbě je množina vstupních adjungovaných boxů minimální (4 boxy) a množina výstupních adjungovaných boxů je také



minimální (4 boxy). Pokud volíme permutace pečlivěji, můžeme docílit toho, že adjungovaných vstupních boxů bude 13 (maximum) a počet adjungovaných výstupních boxů bude 16 (maximum), viz příklad na obrázku 6.



Obr.6: Náhodná volba permutací, adjungované boxy

**Závěr.** Permutace typu Small-Middle-Large  $\mathbf{SMLPerm}_{i,j}$  na množině  $\{0, \dots, c - 1\}$  můžeme volit libovolně, pouze musí zajišťovat maximalitu odpovídající difúzní úrovni. Pravděpodobně je lepší je volit náhodně nebo co nejvíce nepravidelně a zajistit dostatečný počet adjungovaných boxů.

### 3.8 Konstanty $\mathbf{RConstB}_{i,j}$

Konstanty  $\mathbf{RConstB}_{i,j}$  o  $c$  bajtech,  $i = 0, \dots, \rho - 1$ ,  $j = 0, \dots, r - 1$  mají za cíl odlišit jednotlivé transformace  $T_1$ . Lze je zahrnout do definice  $S$ -boxů, neboť způsobují pouze jejich posun o konstantu (viz důkaz pro konstanty  $\mathbf{RConstF}_{i,t}$  ve funkci  $\Phi$  níže). V případě, že v celé funkci  $\Pi$  je použit pouze jeden  $S$ -box (to může být vhodné u některých HW realizací), rundovní konstanty definují až 256 variant jeho posunu o konstantu. V tom případě je ideální náhodná

volba konstant  $\mathbf{RConstB}_{i,j}$  o  $c$  bajtech,  $i = 0, \dots, \rho - 1, j = 0, \dots, r - 1$ . Pokud jsou všechny S-boxy voleny náhodně, je možné tyto konstanty volit nulové.

## 4. Konstrukce sítě $\Phi$

### 4.1 S-boxy $\text{SubsF}_{i,j,t}$

U rodiny funkcí DN musí být zaručeno, že S-boxy použité ve funkci  $\Phi$  a S-boxy použité ve funkci  $\Pi$  jsou navzájem různé. Ideální je, pokud se liší náhodně. To je opatření proti algebraickým útokům. Cílem je, aby rovnice, které charakterizují funkce  $\Phi$  a  $\Pi$ , používaly různé S - boxy.

Žádný z použitých S-boxů (ve funkci  $\Phi$  i ve funkci  $\Pi$ ) by neměl mít algebraické vlastnosti (například jako algebraický S-box AES). To je opatření proti algebraickým útokům, zjednodušujícím zápis vztahů ve funkcích  $\Phi$  a  $\Pi$ .

Není zakázáno použít pouze jeden S-box ve funkci  $\Phi$ , ale ideální volbou jsou náhodně generované S-boxy, které mají vyhovující odolnost proti diferenciální a lineární kryptoanalýze.

Označme  $p_\Phi (q_\Phi)$  maximum z hodnot  $DP^S (LP^S)$  přes všechny S-boxy  $\text{SubsF}_{i,j,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1, j = 0, \dots, r - 1$ ), použité ve funkci  $\Phi$ . Počet velkých rund DN je závislý na velikosti hodnot  $p_\Phi$  a  $q_\Phi$ . Čím jsou menší, tím méně velkých rund  $\rho$  může DN mít (viz dále).

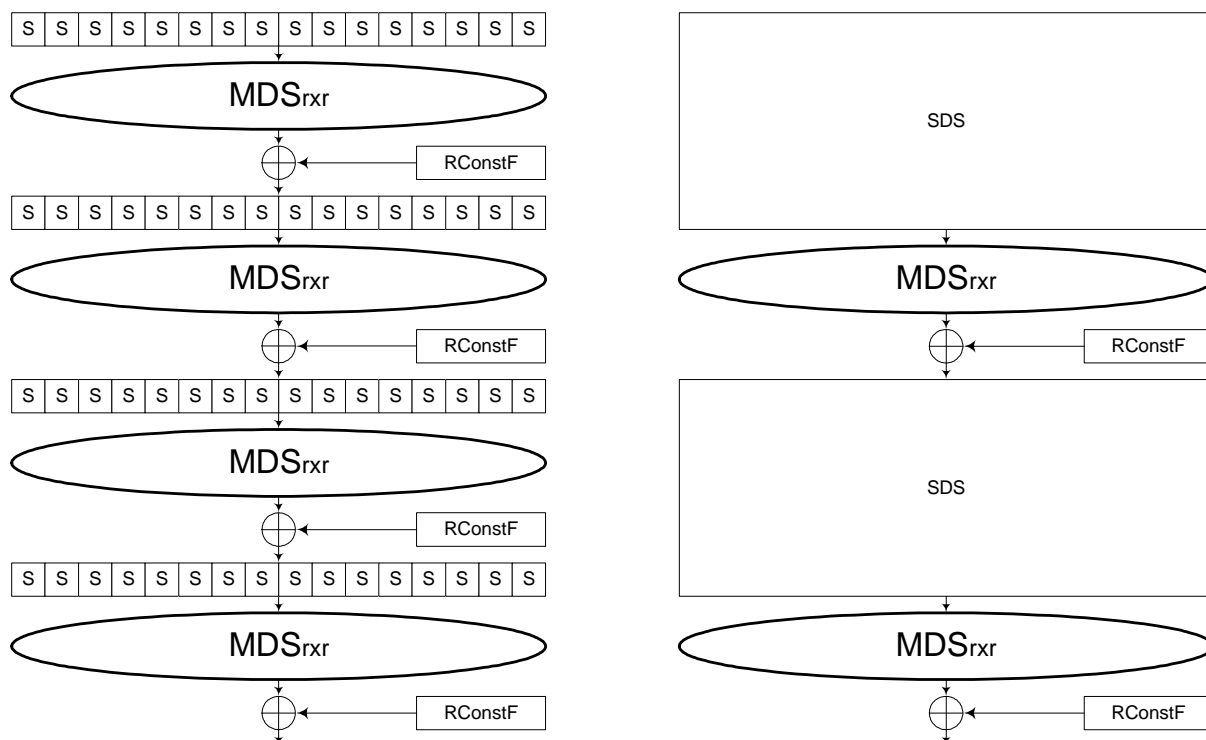
Ideální volba jsou náhodné nebo pseudonáhodné S-boxy, které mají dostatečnou odolnost proti lineární a diferenciální kryptoanalýze. Na velikosti čísel  $p_\Phi (q_\Phi)$  závisí odolnost sítě  $\Phi$  proti DC a LC a volba počtu velkých rund, viz dále.

### 4.2 $\Phi$ jako systém blokových šifer $F_t$

Ve funkci  $\Phi$  jsou volitelné S-boxy  $\text{SubsF}_{i,j,t}$  ( $i = 1, \dots, \rho - 1, j = 0, \dots, r - 1, t = 0, \dots, c - 1$ ), matice  $\mathbf{MDS}_{i,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1$ ) a konstanty  $\mathbf{RConstF}_{i,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1$ ). Cílem volby těchto prvků je, aby se všechny sloupcové transformace  $F_t$  ( $t = 0, \dots, c - 1$ ) co nejvíce lišily (nejlépe náhodně), byly co nejvíce náhodně voleny a byly co nejvíce odolné proti diferenciální a lineární kryptoanalýze. Náhodná volba těchto prvků způsobí velkou paměťovou náročnost DN. Proto minimálním požadavkem je, aby všechny transformace  $F_t$  ( $t = 0, \dots, c - 1$ ) byly různé a odolné proti diferenciální a lineární kryptoanalýze.

### 4.3 Odolnost transformací $F_t$ proti DC a LC

Každá sloupcová transformace  $F_t$ ,  $t = 0, \dots, c - 1$ , je součinnou blokovou šifrou  $F_t = f_{\rho-1,t} \bullet \dots \bullet f_{2,t} \bullet f_{1,t}$  s délkou bloku  $r$  bajtů. Můžeme ji také vyjádřit jako součin  $\rho/2$  SDS sítí, které jsou spojeny difúzní úrovní.



Obr. 7: Sloupcová transformace jako součinná šifra, jejíž runda je SDS síť

Chápeme-li jednu SDS síť jako jednu rundu blokove šifry  $F_t$ , můžeme maximální diferenciální pravděpodobnost  $DP^{SDS}$  a maximální lineární pravděpodobnost  $LP^{SDS}$  každé této rundy odhadnout podle Věty 1 a Věty 2 (viz Dodatek A).

**Věta 5. Odolnost blokove šifry  $F_t$ ,  $t = 0, \dots, c - 1$ , proti DC a LC.**

Spojením dvou následujících rund blokove šifry  $F_t = f_{p-1,t} \cdot \dots \cdot f_{2,t} \cdot f_{1,t}$  vzniká SDS síť, pro níž platí

$$DP^{SDS} \leq (p_\Phi)^r,$$

$$LP^{SDS} \leq (q_\Phi)^r.$$

**Důkaz.** Vyplývá přímo z Věty 1 a Věty 2 (Dodatek A).

Z nedostatku důkazových metod nemáme jinou možnost, než odolnost  $F_t$  proti DC a LC měřit čísly  $DP^{SDS}$  a  $LP^{SDS}$ .

**Důsledek 1. Současný nejlepší odhad odolnosti  $F_t$  proti DC.** Poznamenejme, že podle [NyKn92] lze k odhadu  $DP^{F_t}$  pro  $F_t = SDS \cdot \dots \cdot SDS \cdot SDS$  použít  $DP^{F_t} \leq (DP^{SDS})^2 \leq (p_\Phi)^{2r}$  pokud jsou použity alespoň 4 sítě SDS, tj. **8 substitučních úrovní (8 velkých rund)**. Hodnota  $DP^{F_t}$  je pravděpodobně nižší než uvedený (nejlepší současný) odhad  $DP^{SDS} \times DP^{SDS} \leq (p_\Phi)^{2r}$ , ale zatím chybí důkazové metody.

**Důsledek 2. Současný nejlepší odhad odolnosti  $F_t$  proti LC.** U odhadu odolnosti  $F_t$  proti LC můžeme vycházet pouze z toho, že máme odhad  $LP^{SDS} \leq (q_\Phi)^r$  jako jedné "rundy" součinné šifry  $F_t = SDS \cdot \dots \cdot SDS \cdot SDS$ .

### 4.3.1 Poznámka k odolnosti transformací $F_t$ proti DC a LC

Je požadováno, aby všechny transformace  $F_t$  ( $t = 0, \dots, c - 1$ ) byly co možná nejvíce odolné proti diferenciální a lineární kryptoanalýze. Klasická lineární a diferenciální kryptoanalýza není v případě funkcí  $F_t$  přímo využitelná, protože  $F_t$  je bloková šifra s konstantními rundovními klíči. Opatřeními proti lineární a diferenciální kryptoanalýze ve skutečnosti zajišťujeme to, aby mezi vstupy a výstupy funkce  $F_t$  (a eventuelně i mezi mezivýstupy v jednotlivých rundách) neexistovaly využitelné lineární nebo diferenční vztahy. Na tyto vlastnosti má největší vliv volba S-boxů a velikost hodnot  $p_\Phi$  a  $q_\Phi$ .

### 4.4 Matice $MDS_{i,t}$ a maximalita difúzní úrovně $F_t$

U rodiny funkcí DN musí být zaručeno, že matice  $MDS_{i,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1$ ) v transformacích  $F_t$  ( $t = 0, \dots, c - 1$ ), tj. ve funkci  $\Phi$ , jsou typu MDS (maximum distance separable). Měly by dále zajistit difúzi na úrovni bitů a nikoli bajtů jako celku, což například nesplňuje matice MDS, obsahující pouze prvky (hex.) 0x00 a (hex.) 0x01. Těchto prvků by matice MDS měly obsahovat zcela minimální množství. V binárním vyjádření  $8r \times 8r$  by matice MDS neměla být ani příliš řídkou ani příliš pravidelnou. Měla by být co nejvíce náhodnou binární maticí o rozměru  $8r \times 8r$ . Ideální je, pokud všechny matice  $MDS_{i,t}$  jsou různé a vytvořeny náhodně. To je opatření proti algebraickým útokům. Není však striktně zakázáno použít všechny matice stejné.

### 4.5 Konstanty $RConstF_{i,t}$

Konstanty  $RConstF_{i,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1$ ) mají v definici DN pouze metodický význam. Lze je zahrnout do definice S-boxů, neboť způsobují pouze jejich posun o konstantu (viz poznámka níže). V případě, že v celé funkci  $\Phi$  je použit pouze jeden S-box (to může být vhodné u některých HW realizací), rundovní konstanty definují až 256 variant jeho posunu o konstantu. V tom případě je ideální náhodná volba konstant  $RConstF_{i,t}$  ( $i = 1, \dots, \rho - 1, t = 0, \dots, c - 1$ ). Pokud jsou všechny S-boxy voleny náhodně, je možné tyto konstanty volit nulové.

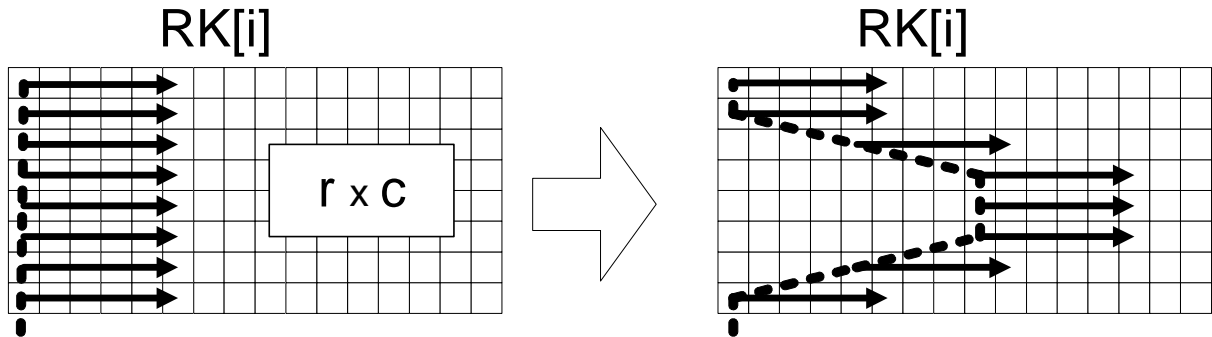
**Poznámka. Převod rundovních konstant do S-boxů.** Rundovní konstantu lze triviálně převést na posun S-boxů o konstantu. Označme posunutý S-box jako  $SubsF_{i,j,t}^*(x) = SubsF_{i,j,t}(x) \oplus a_{i,j,t}$ . Posun vypočteme jako  $(a_{i,0,t}, a_{i,1,t}, \dots, a_{i,r-1,t})^T = MDS_{i,t}^{-1} \cdot (RConstF_{i,0,t}, RConstF_{i,1,t}, \dots, RConstF_{i,r-1,t})^T$ . Máme

$$MDS_{i,t} \cdot (SubsF_{i,0,t}^*(RK_{i-1,0,t}), SubsF_{i,1,t}^*(RK_{i-1,1,t}), \dots, SubsF_{i,r-1,t}^*(RK_{i-1,r-1,t}))^T \oplus (0, 0, \dots, 0)^T = MDS_{i,t} \cdot (SubsF_{i,0,t}(RK_{i-1,0,t}) \oplus a_{i,0,t}, SubsF_{i,1,t}(RK_{i-1,1,t}) \oplus a_{i,1,t}, \dots, SubsF_{i,r-1,t}(RK_{i-1,r-1,t}) \oplus a_{i,r-1,t})^T = MDS_{i,t} \cdot (SubsF_{i,0,t}(RK_{i-1,0,t}), SubsF_{i,1,t}(RK_{i-1,1,t}), \dots, SubsF_{i,r-1,t}(RK_{i-1,r-1,t}))^T \oplus MDS_{i,t} \cdot (a_{i,0,t}, a_{i,1,t}, \dots, a_{i,r-1,t})^T = MDS_{i,t} \cdot (SubsF_{i,0,t}(RK_{i-1,0,t}), SubsF_{i,1,t}(RK_{i-1,1,t}), \dots, SubsF_{i,r-1,t}(RK_{i-1,r-1,t}))^T \oplus (RConstF_{i,0,t}, RConstF_{i,1,t}, \dots, RConstF_{i,r-1,t})^T = (RK_{i,0,t}, RK_{i,1,t}, \dots, RK_{i,r-1,t})^T, \text{ q.e.d.}$$

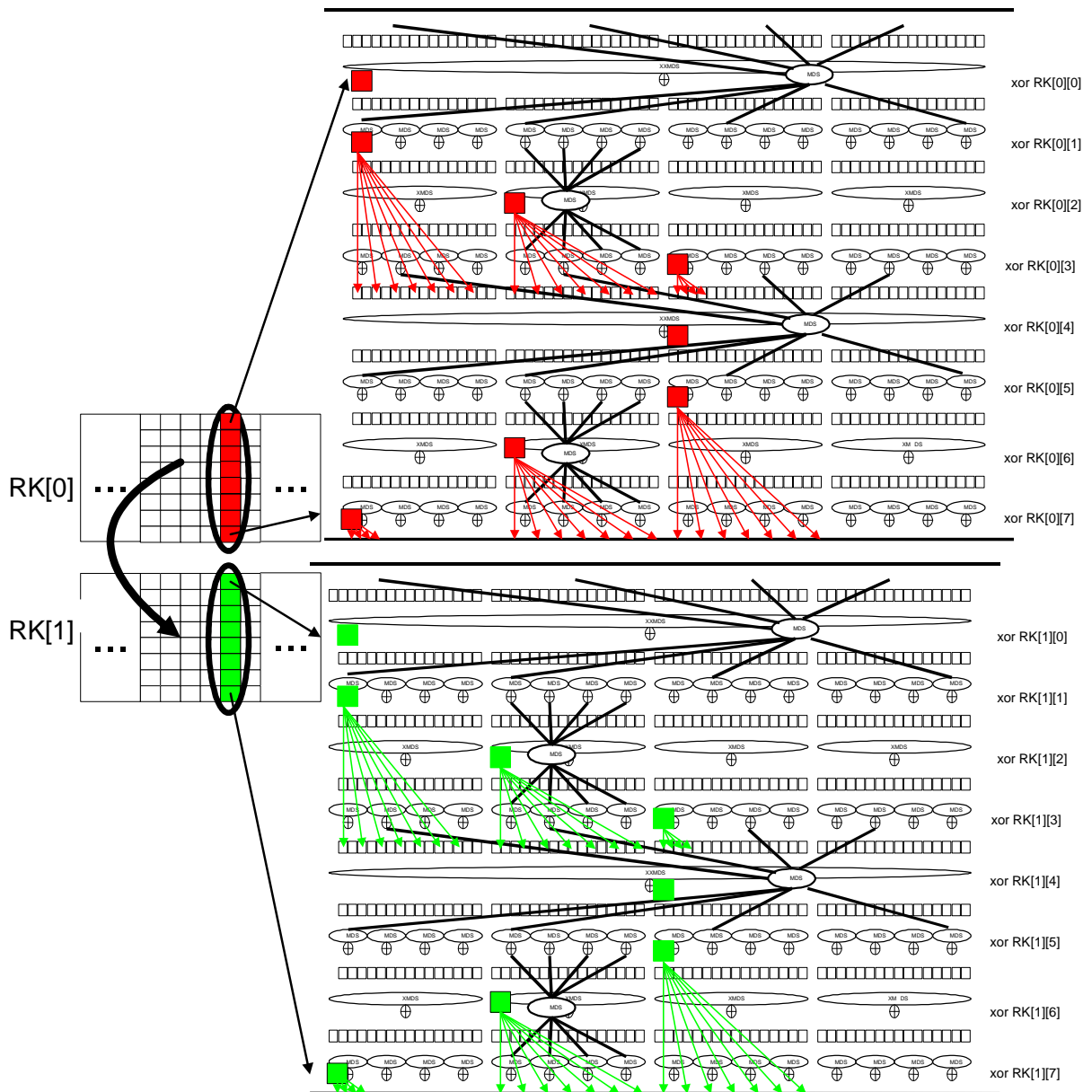
### 4.6 Závěrečná klíčová permutace KeyPerm

Ve funkci  $\Phi$  je volitelná závěrečná klíčová permutace KeyPerm. Tato permutace není z bezpečnostního hlediska povinná, jejím cílem je zefektivnit difúzi rundovních klíčů uvnitř funkce  $\Pi$ . Protože změny v poli RK se šíří zvláště ve sloupcích, cílem KeyPerm je změny v jednom sloupci pole rundovních klíčů promítnout do co největšího počtu různých boxů funkce  $\Pi$ . KeyPerm může být velmi jednoduchou permutací, například pouze cyklicky posune některé řádky v poli RK například takto:  $RK: RK[i][j][k] = RK[i][j][(k + \text{shift\_row\_}j)$

mod  $c$ ], viz obr. 8. Konkrétní definice KeyPerm závisí na konkrétní struktuře funkce  $\Pi$ , jak ukazuje obr. 9. Na obr. 9 například vidíme, že KeyPerm nemá smysl u těch rundovních klíčů, které se načítají po aplikaci (největší) matice XXXMDS. Tam promíchávání mezi největšími boxy zajišťuje sama matice.



Obr. 8: Příklad KeyPerm



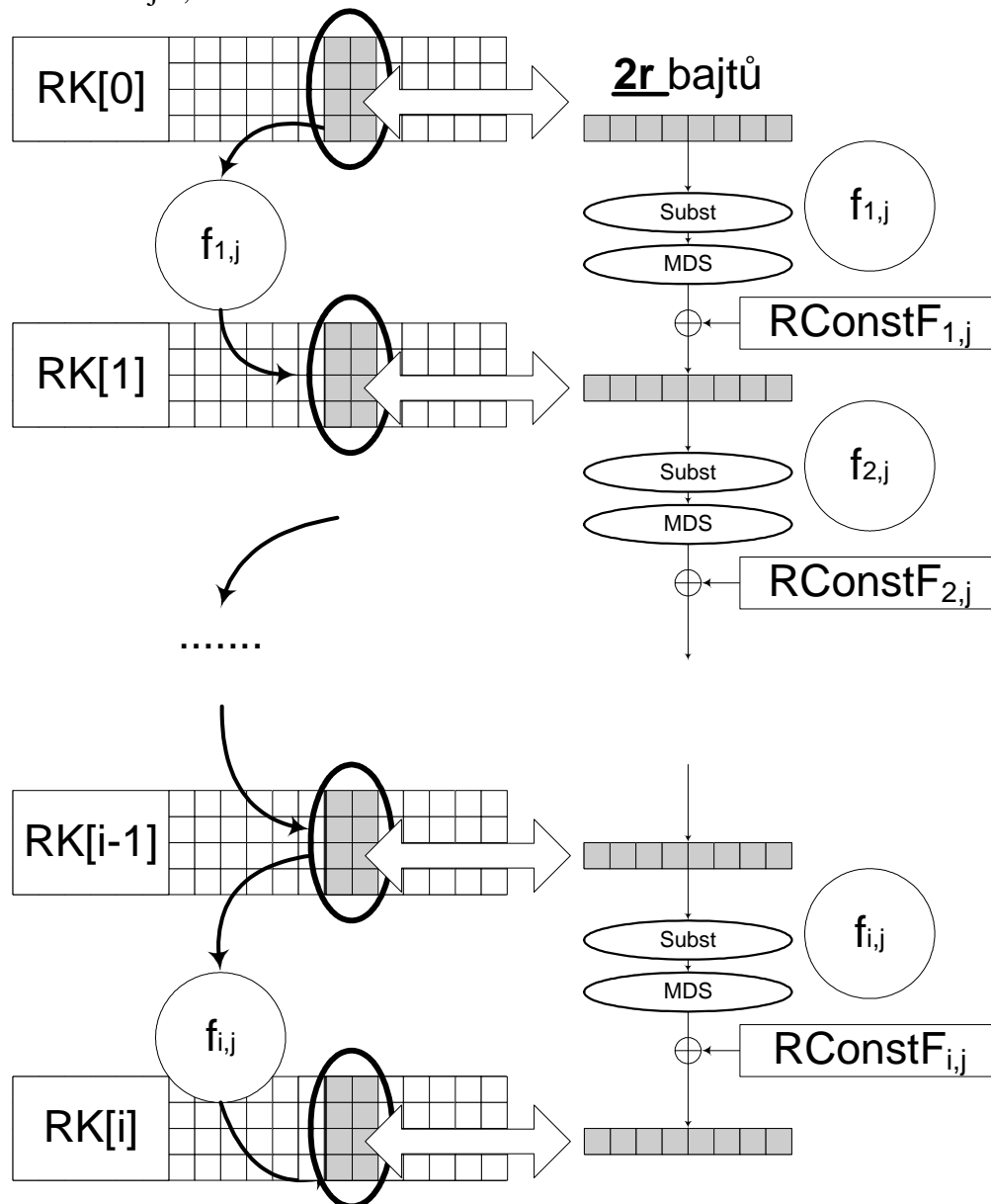
Obr.9: Příklad difúze s využitím závěrečné klíčové permutace

## 5. Double Net jako zesílený šifrovací algoritmus

DN byla konstruována pro použití s konstantním otevřeným textem jako náhodné orákulum v hašovací funkci ([K106]). V tomto případě se nazývá speciální bloková šifra.

Pokud budeme DN uvažovat s proměnným otevřeným textem, může být použita jako algoritmus pro šifrování. V tom případě má oproti klasickým blokovým šifrám výhodu silného zpracování klíče, což ji činí více odolnou proti budoucím útokům ze strany klíče.

DN jako algoritmus pro šifrování nebude mít obvykle tak dlouhý klíč jako v případě hašování, tj. pole  $r \times c$  může být relativně malé a rozměr  $c$  (šířka otevřeného textu v bajtech) může být také relativně malý. Klasickým příkladem může být 128 bitová bloková šifra s 256 bitovým klíčem, tj.  $c = 16$  a  $r = 2$ . Principy sloupcové transformace lze zachovat i tehdy, když sdružíme několik sousedních sloupců dohromady (na obrázku 10 jsou to dva sloupce) a chápeme je jako jeden "silnější" sloupec, na nějž aplikujeme sloupcovou transformaci  $F$  o délce  $2r$  bajtů, viz obr. 10.



Obr. 10: Princip sloupcové transformace, aplikovaný na několik sloupců

## 6. Počet rund DN, varianty a rychlost hašování

### 6.1 Počet rund: 6 (10)

Kvalita substitučních boxů a rozměry pole rundovních klíčů použitých ve funkci  $\Phi$  určí vztah mezi počtem rund a odhady odolnosti  $\Phi$  proti DC a LC. Podobné odhady pro funkci  $\Pi$  zajistíme mnohem snadněji. Pro stanovení odolnosti počtu rund ve funkci  $\Phi$  je také důležité, zda se DN použije pro hašování nebo pro šifrování. U šifrování mají odhady větší důležitost, u hašování nejsou metody LC a DC přímo využitelné klasickým způsobem, neboť klíč je známou konstantou. Počet rund závisí také na velikosti bezpečnostní rezervy. U funkce DN(512, 8182) jsme například stanovili počet rund 10 se stávajícími S-boxy. Pokud budou voleny S-boxy kvalitnější, může být počet rund snížen na 6.

### 6.2 Varianty DN

Podstatou sítě DN je, že klíče  $a, b, \dots, z$  dílčích šifer součinné šifry  $\Pi = B_z \cdot \dots \cdot B_b \cdot B_a$  jsou samy vytvářeny kvalitní blokovou šifrou  $\Phi$ . Se zvyšováním počtu rund se klíče ( $a, b, \dots$ ) a ( $\dots, y, z$ ) stávají nezávislými (náhodnými veličinami) neboť odpovídají vztahu otevřeného a šifrovaného textu blokové šifry  $\Phi$ . Potom i blokované šifry ( $B_a, B_b, \dots$ ) a ( $\dots, B_y, B_z$ ) se stávají nezávislými (náhodnými) blokovými šiframi. Promíchání sloupců pole RK mezi sebou a s otevřeným textem zajistí funkce  $\Pi$ . Odolnost funkce  $\Pi = B_z \cdot \dots \cdot B_b \cdot B_a$  proti DC a LC bude obvykle zajištěna už při součinu několika velkých rund  $B$ . Proto můžeme v součinu  $\Pi = B_z \cdot \dots \cdot B_b \cdot B_a$  vynechat střední část a použít jen několik počátečních a několik koncových blokovaných šifer  $B$ , například tři a tři ( $\Pi = B_z \cdot B_y \cdot B_x \cdot B_c \cdot B_b \cdot B_a$ ).

### 6.3 Rychlost hašování

Zde srovnáme rychlost hašovacích funkcí HDN(512, 8192) s SHA-256 a SHA-512 a Whirlpool. Tyto algoritmy jsou obsaženy ve veřejně dostupné knihovně Crypto++. Autorem je Wei Dai a zdrojové kódy i testy rychlosti lze nalézt na <http://www.eskimo.com/~weidai/benchmarks.html>. Všechny algoritmy v knihovně Crypto++ byly napsány v C++, kompilovány pomocí Microsoft Visual C++.NET 2003 (optimalizace celého programu na rychlost) a spuštěny na Pentium 4 (2.1 GHz) pod Windows XP SP 1. V následující tabulce uvádíme jejich vzájemné rychlosti a v další části tabulky pak rychlosti naší vlastní implementace SHA-256, SHA-512 a HDN(512, 8192). Uvedené testy vlastní implementace probíhaly na notebooku Acer (Pentium, 1.6 GHz) v OS Windows XP SP2, kompilace pod MS Visual C++ 6.0.

	knihovna Crypto++	Pentium 4 (2.1 GHz)	
Algoritmus	Testováno megabajtů	rychlost v MByte/s	
MD5	1002	216	
SHA-1	256	68	
SHA-256	256	<b>44</b>	
SHA-512	64	<b>11</b>	
Whirlpool	64	<b>12</b>	
	Vlastní implementace	Pentium, 1.6 GHz	

Algoritmus	Testováno megabajtů	rychlost v MByte/s	Varianta algoritmu HDN "tři + tři" $\Pi = B_z \cdot B_y \cdot B_x \cdot B_c \cdot B_b \cdot B_a$
SHA-256	64	32	
SHA-512	64	<b>17</b>	
HDN(512, 8192)-1	64	136	
HDN(512, 8192)-2	64	35	
HDN(512, 8192)-3	64	20	20.48
HDN(512, 8192)-4	64	14	15.70
HDN(512, 8192)-5	64	11	12.78
<b>HDN(512, 8192)-6</b>	64	<b>9.09</b>	<b>10.75</b>
HDN(512, 8192)-7	64	7.67	9.28
HDN(512, 8192)-8	64	6.65	8.15
HDN(512, 8192)-9	64	5.84	7.30
<b>HDN(512, 8192)-10</b>	64	<b>5.22</b>	<b>6.57</b>

Tab.: Porovnání rychlostí hašovacích algoritmů

Údaje v tabulce je třeba brát orientačně, neboť rychlost hašování velmi záleží na zvolené metodě a různých optimalizacích. Orientačně lze říci, že rychlost hašování HDN(512, 8192)-10 je třikrát pomalejší než SHA-512 (a Whirlpool) a rychlost hašování HDN(512, 8192)-6 je dvakrát pomalejší než SHA-512.

Protože 10 velkých rund jsme u HDN(512, 8192) zvolili jen pro zajištění odolnosti funkce  $\Phi$  proti DC a LC a u funkce  $\Pi$  byla odolnost zajištěna i s rezervou již při 6 velkých rundách, můžeme v tomto případě použít také variantu "tři a tři", tj.  $\Pi = B_z \cdot B_y \cdot B_x \cdot B_c \cdot B_b \cdot B_a$ .

Výsledky měření ukazují, že HDN(512, 8192) není jen teoretická konstrukce, ale zcela prakticky využitelná hašovací funkce, která je jen 2-3krát pomalejší než SHA-512 a Whirlpool.

## 7. Závěr

V tomto příspěvku popisujeme rodinu speciálních blokových šifer DN a rodinu hašovacích funkcí HDN podle koncepce SNMAC [Kl06]. Ukazuje se, že to není jen teoretický koncept, ale použitelné funkce, jejichž rychlost je jen 2-3krát nižší než rychlost SHA-512 a Whirlpool.

U hašovací funkce má útočník možnost manipulovat všemi vstupy, zatímco klasická bloková šifra byla konstruována za předpokladu, že obsahuje nějaký prvek, který útočník nezná a neovládá (šifrovací klíč). Speciální blokovou šifru jsme museli konstruovat za předpokladu, že útočník její šifrovací klíč zná a že má možnost s ním libovolně manipulovat.

Základní myšlenka speciální blokové šifry DN je jednoduchá – oproti klasické blokové šifře věnuje zpracování klíče stejnou pozornost jako klasická bloková šifra věnovala zpracování otevřeného textu. Jedna SP síť tedy zajišťuje míchání klíče a druhá míchání otevřeného textu s klíčem.



Zároveň uvádíme myšlenku, že i klasické blokové šifry jako primitiva, určená k šifrování dat, by se měly v budoucnu konstruovat podobně jako speciální blokové šifry. O klíči se předpokládalo, že je neznámý útočníkovi a že s ním útočník nemůže manipulovat. Moderní technologie a rostoucí možnosti útočníků však ukazují, že tyto předpoklady nebudou stále více odpovídat skutečnosti. Jsou to útoky, které již částečně známe – útoky postranními kanály, útoky příbuznými klíči, pravoúhelníkové útoky apod. (viz například [Bi93], [Bi03], [Ki04], [Ho05], [Ki05], [Bi05], [Bi06]) a další útoky, které budou jistě teprve objeveny v dalších desetiletích. Všechny mají společné to, že původní předpoklad o neznalosti klíče protivníkem nebo o nemožnosti s ním manipulovat nějakým způsobem oslabují. Tyto útoky budou vznikat stále častěji s rozšiřováním kryptografických metod a prostředků, což dokládá vývoj v několika posledních desetiletích. Nová generace blokových šifer by proto měla být odolná i proti útokům ze strany klíče. Je otázkou dalšího výzkumu, zda speciální blokové šifry jsou tím správným východiskem. V každém případě by funkce zpracovávající klíč měly být v moderních blokových šifrách posíleny.

## Dodatky

V příštím e-zinu Crypto-World 4/2007 budou uvedeny následující dodatky k této práci:

Dodatek A: Teorie SP sítí a jejich odolnost proti DC a LC

Dodatek B: Definice volitelných prvků speciální blokové šifry DN(512,8192)

Dodatek C: Popis volitelných prvků HDN(512, 8192)

Dodatek D: Zdrojové kódy DN(512, 8192) a HDN(512, 8192)

Dodatek E: Testovací hodnoty DN(512, 8192) a HDN(512, 8192)

## Poděkování

Autor děkuje Tomáši Rosovi za mnoho užitečných připomínek k předchozím verzím příspěvku.

## 8. Literatura

[Bi93] E. Biham, New Types of Cryptanalytic Attacks Using Related Keys, Proceedings of EUROCRYPT **1993**, pp. 398-409, LNCS 765, 1993.

[Bi03] E. Biham, Orr Dunkelman, and Nathan Keller, Rectangle Attacks on 49-Round SHACAL-1, FSE **2003**, LNCS 2887, p. 22 ff.

[Bi94] E. Biham, *On Matsui's Linear Cryptanalysis*, Advanced in cryptology-EUROCRYPT'94, pp. 341-355, Springer-Verlag, **1994**.

[BiSh91a] E. Biham and A. Shamir, *Differential Cryptanalysis of DES-like Cryptosystem*, Journal of Cryptology, Vol.4, pp. 3-72, **1991**.

[BiSh91b] E. Biham and A. Shamir, *Differential Cryptanalysis of Snefru, Khafre, REDOC-II, LOKI and Lucifer*, Advanced in cryptology-CRYPTO'91, pp. 156-171, Springer-Verlag, **1991**.

[Bi05] Eli Biham, Orr Dunkelman, Nathan Keller: Related-Key Boomerang and Rectangle Attacks, EUROCRYPT **2005**, LNCS 3494, pp. 507–525, 2005.

[Bi06] Eli Biham, Orr Dunkelman, Nathan Keller: Related-Key Impossible Differential Attacks on 8-Round AES-192, CT-RSA **2006**, LNCS 3860, pp. 21–33, 2006.

[Da95] J. Daemen, Cipher and hash function design strategies based on linear and differential cryptanalysis, Doctoral Dissertation, March **1995**, K.U. Leuven.

- [Ho00] Seokhie Hong, Sangjin Lee, Jongin Lim, Jaechul Sung, Donghyeon Cheon, and Inho Cho: Provable Security against Differential and Linear Cryptanalysis for the SPN Structure, FSE **2000**, LNCS, Vol. 1978, pp. 273 - 283
- [Ho05] Seokhie Hong, Jongsung Kim, Sangjin Lee, Bart Preneel: Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192, FSE 2005, LNCS 3557, pp. 368–383, **2005**.
- [Chu03] K. Chun, S. Kim, S. Lee, S.H. Sung, and S. Yoon: Differential and linear cryptanalysis for 2-round SPNs, Information Processing Letters, Vol. 87 (**2003**), pp. 277 - 282.
- [Ka01] Ju-Sung Kang, Seokhie Hong, Sangjin Lee, Okyeon Yi, Choonsik Park, and Jongin Lim: Practical and provable security against differential and linear cryptanalysis for substitution-permutation networks. *ETRI Journal*, 23(4):158–167, **2001**.
- [Ki04] J. Kim, G. Kim, S. Hong, S. Lee and D. Hong, The Related-Key Rectangle Attack-Application to SHACAL-1, Proceedings of International Conference on Information Security and Privacy **2004**, LNCS 3108, pp. 123-136, Springer, 2004.
- [Ki05] Kim J. , Biryukov A., Preneel B, Lee S.: On the Security of Encryption Modes of MD4, MD5 and HAVAL, Cryptology ePrint Archive: Report 2005/327, September - October **2005**, ICICS 2005, <http://eprint.iacr.org/2005/327.pdf>
- [Kl06] V. Klíma, A New Concept of Hash Functions SNMAC Using a Special Block Cipher and NMAC/HMAC Constructions, IACR ePrint archive Report **2006/376**, October, 2006, <http://eprint.iacr.org/2006/376.pdf>
- [Kl06a] V. Klíma: Hašovací funkce nové generace SNMAC, Mikulášská kryptobesídka MKB 2006, Praha, Hotel Olympik, 7. – 8. prosinec **2006**, prezentace a text příspěvku viz domácí stránka projektu [http://cryptography.hyperlink.cz/SNMAC/SNMAC\\_CZ.html](http://cryptography.hyperlink.cz/SNMAC/SNMAC_CZ.html).
- [Kl07] V. Klíma: Rodina speciálních blokových šifer DN a hašovacích funkcí nové generace HDN typu SNMAC, IACR ePrint archive Report 2007/050 , February, **2007**, Testy funkcí DN a HDN v jazyce C, zdrojový kód speciální blokové šifry DN a hašovací funkce HDN viz domácí stránka projektu [http://cryptography.hyperlink.cz/SNMAC/SNMAC\\_CZ.html](http://cryptography.hyperlink.cz/SNMAC/SNMAC_CZ.html).
- [LaMa91] X. Lai, J. L. Massey and S. Murphy *Markov Ciphers and Differential Cryptanalysis*, Advances in Cryptology-EUROCRYPT'91, pp 17-38, Springer-Verlag, **1992**.
- [Ma93] M. Matsui, *Linear cryptanalysis method for DES cipher*, Advanced in cryptology-EUROCRYPT' 93, pp. 386-397, Springer-Verlag, **1993**.
- [Ma94] M. Matsui, *The first Experimental cryptanalysis of DES*, Advanced in cryptology-CRYPTO'94, pp. 1-11, Springer-Verlag, **1994**.
- [NyKn92] K. Nyberg and L. R. Knudsen, *Provable security against a differential attack*, Advanced in cryptology-CRYPTO'92, pp. 566-574, Springer-Verlag, **1992**.
- [Ny94] K. Nyberg, *Linear Approximation of block ciphers*, Advanced in cryptology-EUROCRYPT'94, pp. 439-444, Springer-Verlag, **1994**.
- [PIDi05] James S. Plank, Ying Ding: "Note: Correction to the 1997 tutorial on Reed-Solomon coding", Software: Practice and Experience, Volume 35, Issue 2, pp. 189-194, **2005**, <http://www.cs.utk.edu/~plank/plank/papers/SPE-9-97.html>
- [RiDa97] V. Rijmen, J.Daemen et al, *The cipher SHARK*, Proceedings of the fourth international workshop of Fast Software Encryption, pp. 137-151, Springer-Verlag, **1997**.
- [Ro06] R. M. Roth, Introduction to Coding Theory, Cambridge University Press, **2006**, p. 148.
- [Sa03] F. Sano, K. Ohkuma, H. Shimizu, and S. Kawamura: On the security of nested SPN cipher against the differential and linear cryptanalysis, IEICE Trans. Fundamentals, Vol. E86-A, No.1, January **2003**, pp. 37 - 46.