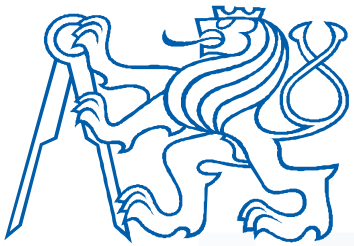
České vysoké učení technické v Praze
Fakulta elektrotechnická
Katedra telekomunikační techniky

A7B32KBE 4. přednáška

Moderní blokové šifry II

Ing. Tomáš Vaněk, Ph.D. tomas.vanek@fel.cvut.cz



Náplň prezentace

- volba AES
- MARS
- RC6
- Serpent
- Blowfish
- Rijndael
- srovnání finalistů

AES

- volba nového algoritmu AES (konec 90. let)
- nástupce DESu
- NIST vyhlásil 2.1.1997 plně veřejnou a otevřenou soutěž s cílem najít silnou šifra pro vládní i komerční použití

Základní požadavky:

- bloková symetrická šifra
- otevřený algoritmus, nechráněný patenty
- bezpečnost algoritmu je důležitější než jeho rychlost

- soutěž trvala čtyři roky

AES – hodnotící kritéria

Bezpečnost	Cena	Algoritmické & implementační charakteristiky
• Skutečná bezpečnost ve srovnání s ostatními soutěžícími algoritmy • Kvalita výstupního ŠT - výstup musí být nerozeznatelný od náhodné permutace stejného vstupního bloku • Bezpečnost algoritmu musí být podložena solidními matematickými základy • Jiné bezpečnostní otázky vzešlé od (odborné) veřejnosti, včetně praktické demonstrace odolnosti vůči kryptoanalytickým útokům	• Licenční požadavky – celosvětově dostupný, neexklusivní licence, poskytovaný zdarma (royalty-free basis) • Licenční požadavky se musí týkat jak HW, tak SW implementace; rychlost algoritmů pod vybranými platformami • Pamětové požadavky – týká se jak HW tak SW implementace; roli hraje např. počet hradel, velikost kódu, požadavky na RAM	• Flexibilita – schopnost pracovat s delšími klíči bloky dat (klíče v rozsahu 128-256bitů po 32bitových krocích, bloky dat po 64 bitech) • Schopnost bezpečné a efektivní implementace v širokém spektru prostředí a aplikací (8bitové procesory, bankomaty, sítě, hlasová a satelitní komunikace, HDTV) • Možnost implementace AES jako proudové šifry, generátoru MAC, generátoru PRN, … • HW a SW přiměřenost • Jednoduchost návrhu

AES - průběh výběrového řízení 1997-2001

červen 1998

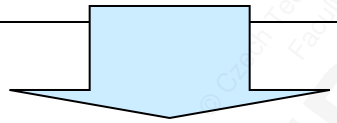
15 kandidátů

USA, Kanada, Belgie, Francie, Německo, Norsko, Velká Británie, Izrael, Jižní Korea, Japonsko, Austrálie a Kostarika

1. Kolo

Bezpečnost
SW efektivita
Flexibilita

červen 1999

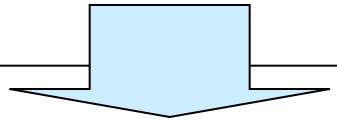


5 finalistů

2. kolo

Mars, RC6, Rijndael, Serpent, Twofish

Bezpečnost
HW efektivita



říjen 2000

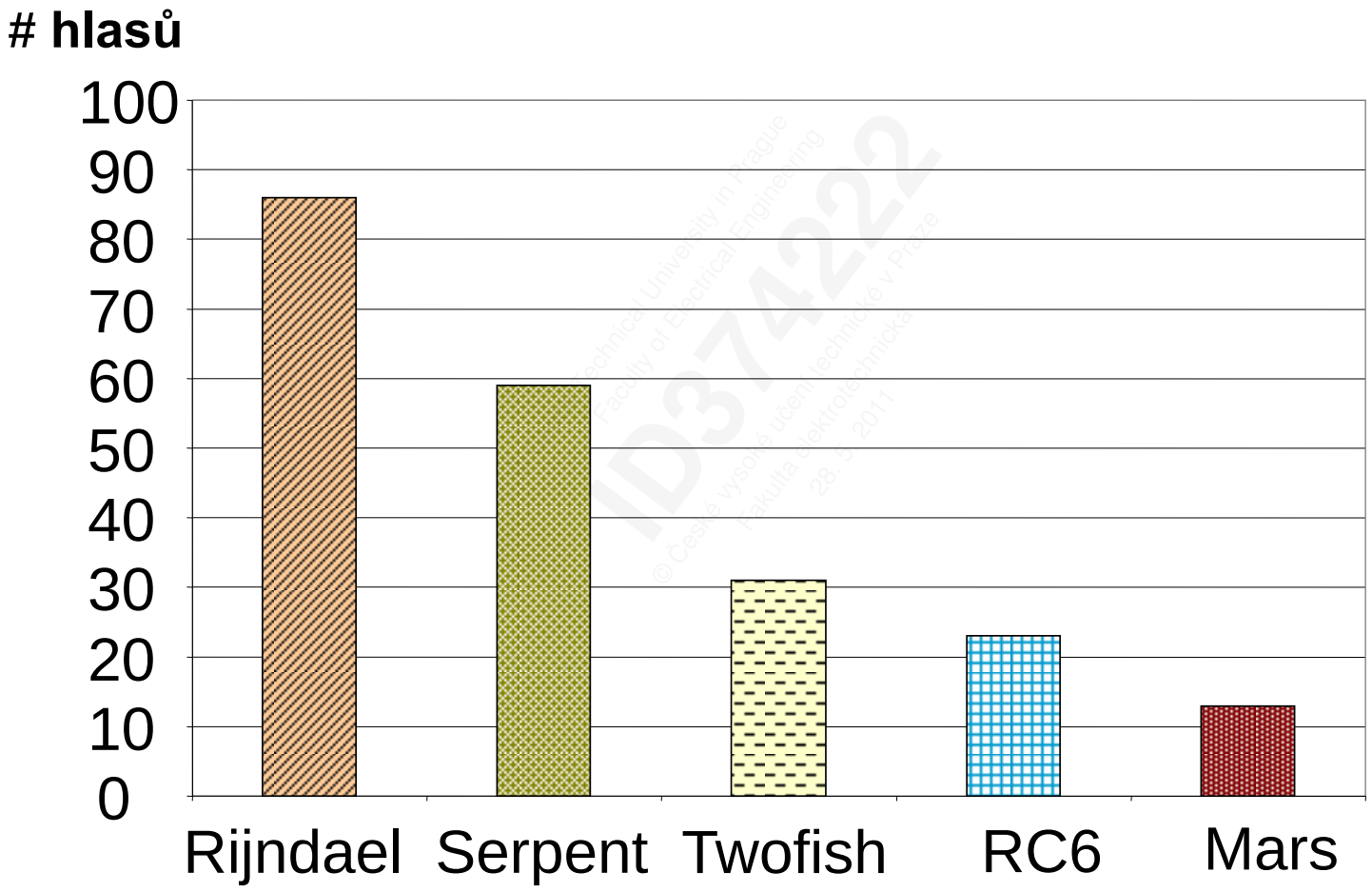
1 vítěz: Rijndael - Belgie

Kandidáti na AES

- CAST-256
- CRYPTON
- DEAL
- DFC
- E2
- FROG
- HPC
- LOKI97
- MAGENTA
- MARS
- RC6
- Rijndael
- SAFER+
- Serpent
- Twofish

	No Response	YES (1)	? (2)	NO (3)	YES - NO	RANK
Rijndael	7	77	19	1	76	1
RC6	4	79	15	6	73	2
Twofish	9	64	28	3	61	3
MARS	5	58	35	6	52	4
Serpent	6	52	39	7	45	5
E2	11	27	53	13	14	6
CAST-256	12	16	58	18	-2	7
SAFER+	13	20	47	24	-4	8
DFC	12	22	43	27	-5	9
Crypton	14	16	43	31	-15	10
DEAL	10	1	22	71	-70	11
HPC	12	1	13	78	-77	12
MAGENTA	9	1	10	84	-83	13
Frog	11	1	6	86	-85	14 (t)
LOKI97	10	1	7	86	-85	14 (t)

Výsledky průzkumu 167 účastníků 3rdAES Conference, Duben 2000



Obecné charakteristiky finalistů

- všech 5 šifer jsou iterované blokové šifry
- všech 5 finalistů používá whiteningu (bělení)
  - promíchání klíče a vstupních/výstupních dat
- 4 finalisté (kromě RC6) používají S-boxy (nelineární substituční funkce)
- 3 finalisté (MARS, Twofish, RC6) používají Feistelovo schéma
- 2 finalisté (Rijndael, Serpent) používají substitučně-permutační sítě (zpracovávají paralelně vstupní blok sérií substitucí a lineárních transformací)

MARS

Autor: IBM

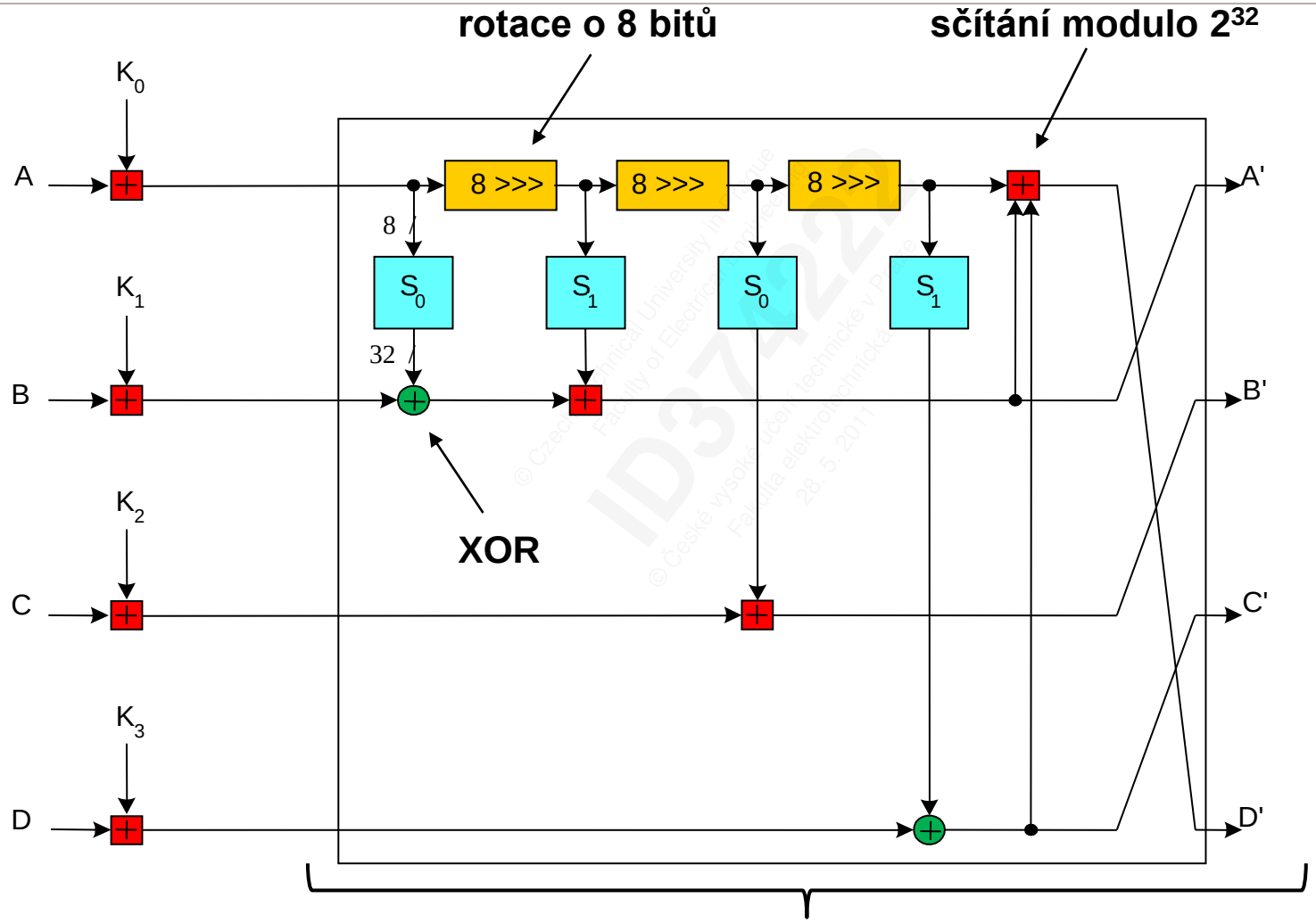
- 5. místo při volbě AES
- velikost bloku 128 bitů
- klíče 128, 192 nebo 256 bitů (dle požadavků NIST)
- obecně podporuje klíče 128 - 448 bitů dlouhé
- využívá rozšířené Feistelovo schéma typu 3
- velmi složitý návrh
  - 16 šifrovacích rund s klíči
  - 16 mixovacích rund bez klíčů

MARS

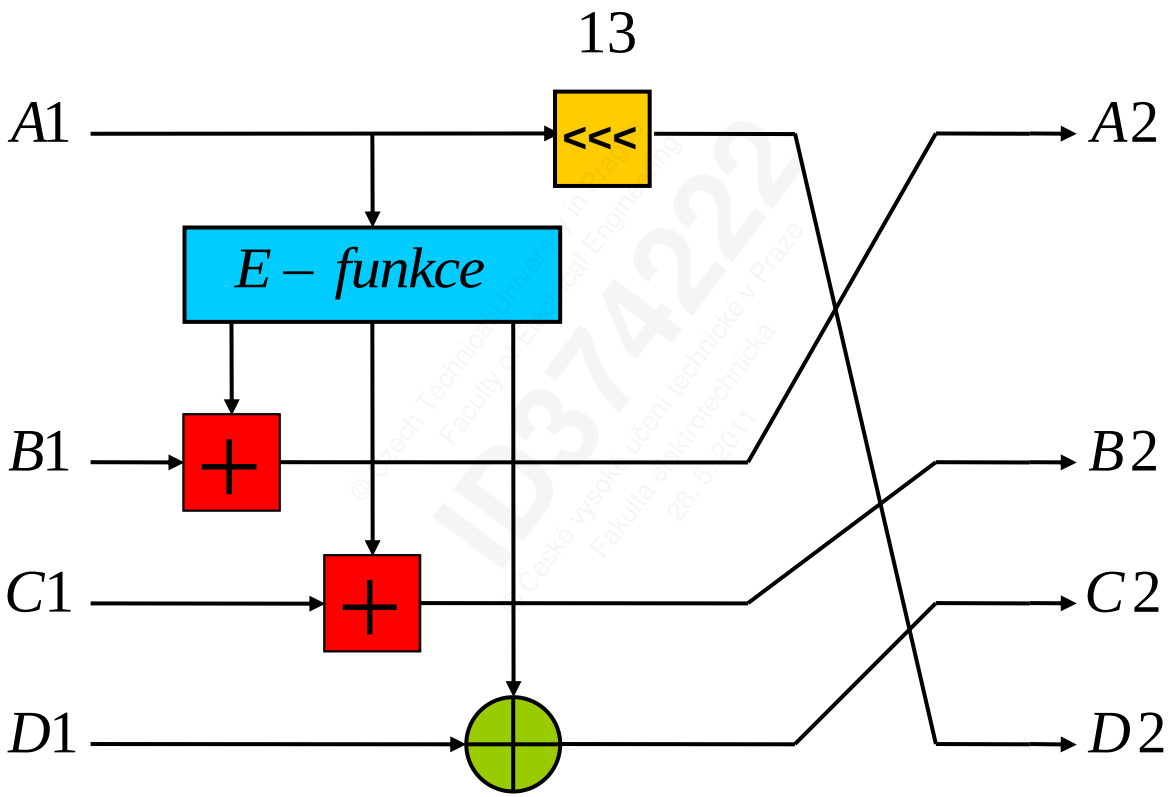
- rundy bez klíčů
  - dva S-boxy
    - každý obsahuje 256 32bitových slov
    - odolné vůči lineární a diferenciální kryptoanalýze
  - sčítání mod 2^{32}
  - XOR
- rundy s klíči
  - násobení v aritmetice mod 2^{32}
  - sčítání v aritmetice mod 2^{32}
  - pevné rotace
  - datově závislé rotace
  - S-box obsahující 512 32bitových slov vzniklý sloučením dvou menších S-boxů (každý s 256 32bitovými slovy)
  - přičítání rundového klíče pomocí XOR

MARS – celkový náhled

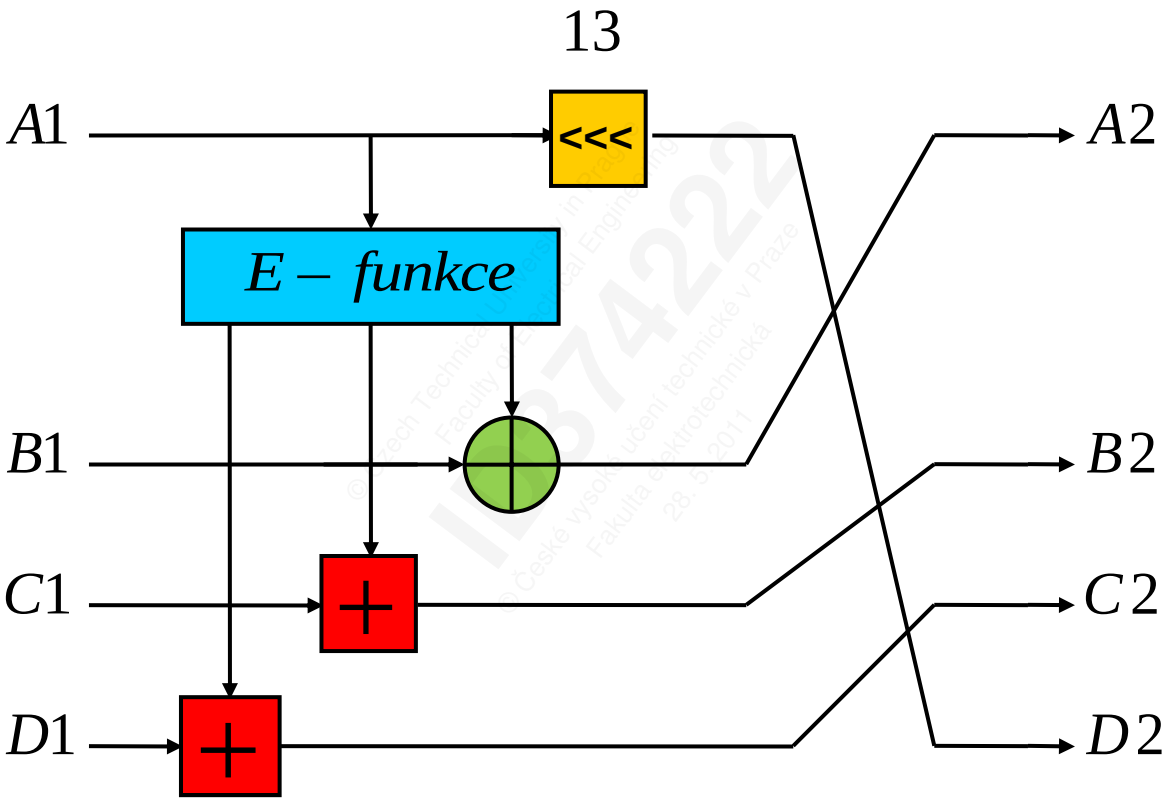
MARS – dopředné mixování



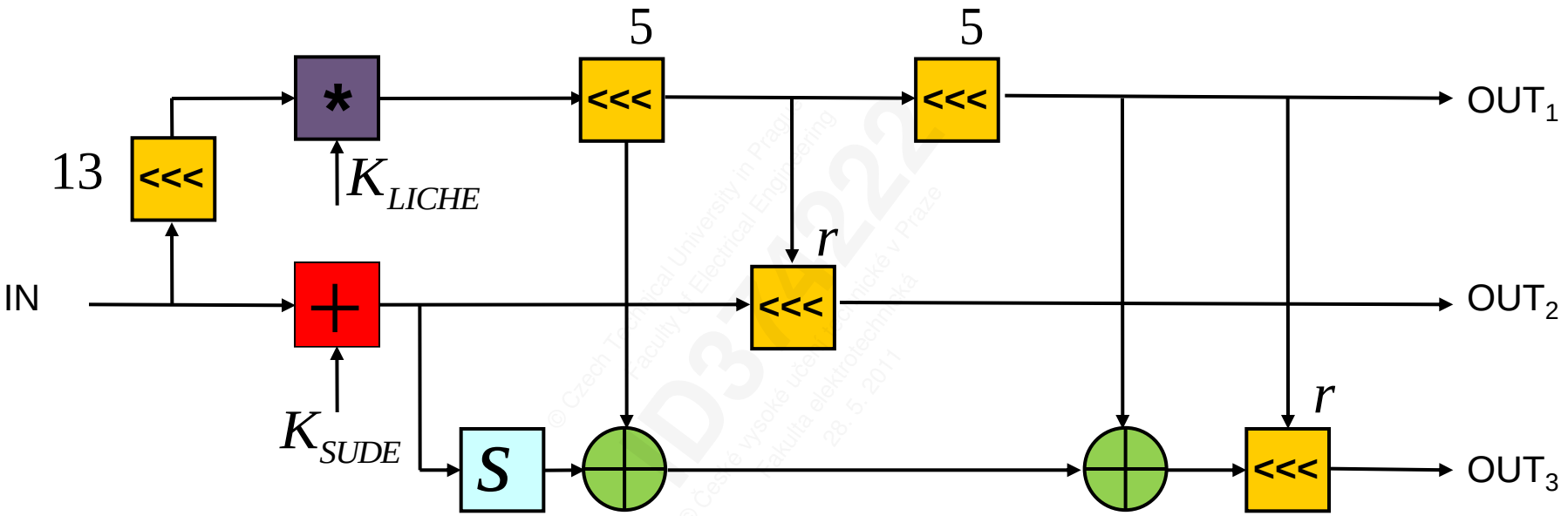
MARS – dopředné šifrování



MARS – zpětné šifrování

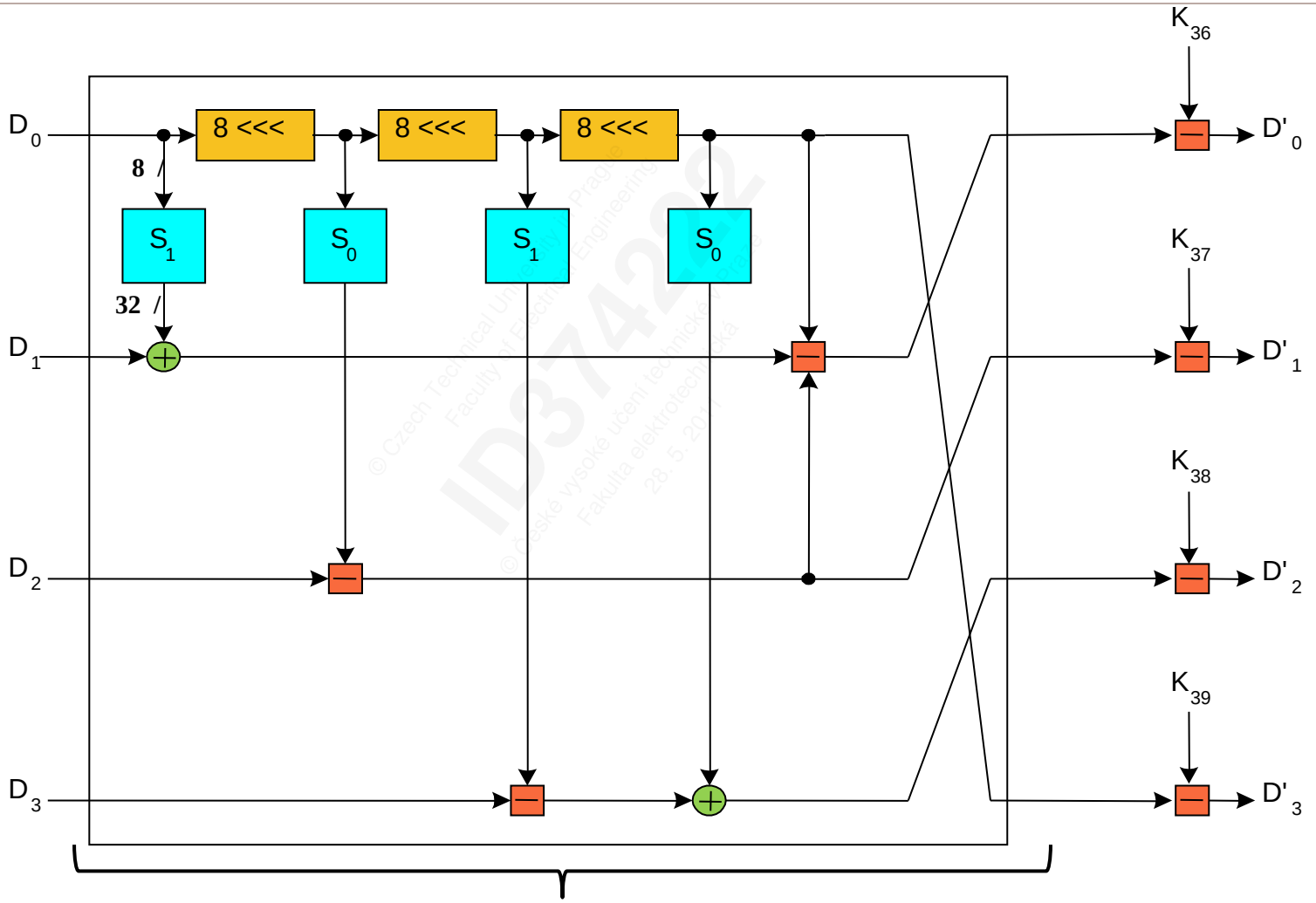


MARS – struktura expanzní E-funkce



- násobení mod 2^{32}
- sčítání mod 2^{32}
- rotace vlevo o 5 a 13 bitů
- rotace vlevo o *r* bitů (r dáno hodnotou pěti posledních bitů)

MARS – zpětné mixování



MARS S-box

- S-box – pole 512 32bitvých čísel
  - někdy se chová jako dva samostatné S-boxy - S_0 S_1
- S-box neobsahuje položku S[i]=(000…0) ani S[i]=(111…1)
- v podboxech S_0 a S_1 se každé dvě položky liší minimálně ve třech ze čtyřech bajtů
- libovolné dvě položky se liší minimálně ve 4 bitech
- Počáteční naplnění $S[5i+j]=SHA\text{-}1(5i,c_1,c_2,c_3)_j$
  - i=0…102 , j=0…4
  - $SHA()_j$ j-té slovo z výstupu SHA-1
  - c_1,c_2,c_3 - desetinný čísla rozvoj čísel π,ε
- v S-boxu neexistuje dvojice S[i],S[j] (i≠j) taková, že:
  - S[i]= S[j]
  - S[i]= ⌐S[j]
  - S[i]= -S[j]

MARS

- dešifrování probíhá zcela stejně jako šifrování (šifra F. typu)

Bezpečnost

- MARS jediný používá dvě nelineární funkce (S-boxy a datově závislé rotace)
- tento fakt spolu s heterogenní strukturou (16 šifrovacích rund a 16 mixovacích rund) zajišťuje větší složitost šifry než u zbývajících kandidátů
- toto je ale zároveň i nevýhoda MARSu
- nejlepší známý útok na MARS předvedl B. Schneier v roce 2000 kdy se mu pomocí útoku se známým OT podařilo prolomit oslabený MARS (pouze 21 z 32 rund z čehož 16 bylo mixovacích (=výrazně jednodušších))

MARS – příprava šifrovacích podklíčů

- celkem je potřeba 40 různých 32bitových podklíčů
- expanze pole $k[]$ obsahujícího n 32bitových slov (n=4...14) na 40 32bitových podklíčů $k_0 \ldots k_{39}$
- expanze klíče mimo jiné zajistí:
  - LSB=1 u klíčů do $K_{liché}$ (viz E-funkce)
  - žádný klíč neobsahuje deset po sobě jdoucích 0 nebo 1
- 3 kroky
  - nakopírování počátečního klíče k do dočasné tabulky T
    - velikost 15 32bitových slov
    - doplněna 0
  - následují čtyři identické části, každá vygeneruje deset 32 bitových slov
    - 4 rundy – promíchání T pomocí Feistelovy sítě typu I
    - volba a další úprava klíče, které se v E-funkci používají k násobení

RC6 - 32/20/16

Autor: RSA Laboratories , jeden ze spoluautorů R.Rivest

- 4. místo
- RC6 má několik volitelných parametrů:
  w - počet bitů slova
  r - počet rund
  b - počet bajtů klíče,

proto se podle nich přesně označuje jako RC6-w/r/b.

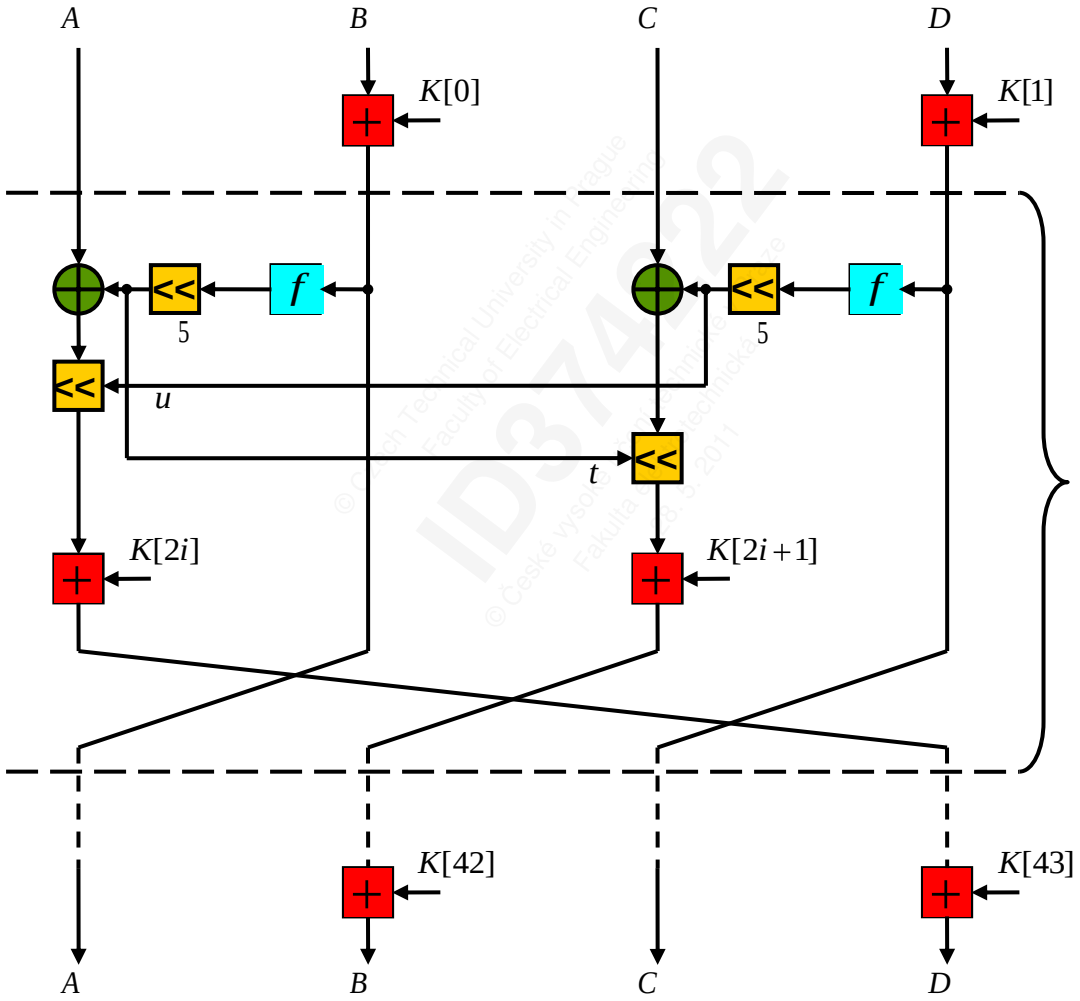
- pro kandidaturu na AES bylo stanoveno
  w = 32b
  r = 20
  b =16B, 24B nebo 32B ,

neboli blok 128 bitů a klíč 128,192 nebo 256 bitů

RC6

- vychází ze starší šifry RC5
  - RC6 = 2 paralelně propojené šifry RC5
- šifra Feistelova typu
- vstup/výstup: 4x32 bitů
- v každé rundě:
  - sčítání mod 2^{32}
  - násobení 2^{32}
    - tato operace není v RC5, zajistí, že rotace bude záviset na každém bitu slova *B* reps. *D* - viz následující slide
  - XOR
  - přičítání klíčů
- nepoužívá S-boxy, ale datově závislé rotace
- existují útoky na zjednodušené verze (15 rund)
- pro deklarovaný počet rund 20 je bezpečný

RC6



$K[x]$ rundový podklíč

$f(x) = x \cdot (2x+1) \bmod 2^{32}$

RC6

Příprava klíče:

- Klíč délky 16, 24 nebo 32 B se uloží do pole a je použit k vytvoření 44 rundových klíčů
- Šifrování i dešifrování jsou si opět velmi podobné. Implementace obou funkcí nevyžaduje více než 10% zdrojů potřebných pro jeden směr.

Bezpečnost

- bez ohledu na jednoduchost designu je RC přiměřeně odolná známým útokům
- není znám žádný útok na 20 rundovou variantu, přestože pro některé autory je počet rund nedostatečnýs
- pomocí lineární a diferenciální krypotanalýzy je možné prolomit 12 rundovou verzi
- statistické útoky založené na vybraných dvojicích OT-ŠT (choosen plain-ciphertext) prokázali zranitelnost až do 13 rund

Serpent

Autoři: Ross Anderson
Eli Biham („objevitel“ diferenciální krypotanalýzy)
Lars Knudsen

- 2.místo
- navržen pro co nejvyšší bezpečnost
- odolný vůči všem dnes známým útokům
- 32 rund (velmi bezpečná, ale pomalá)
- není to šifra Feistelova typu
- substitučně-lineární transformační síť
  - jako Rijndael
- délka bloku 128 bitů (vstup/výstup 4x32 bitů)
- klíč může mát **libovolnou** délku do 256 bitů

Serpent

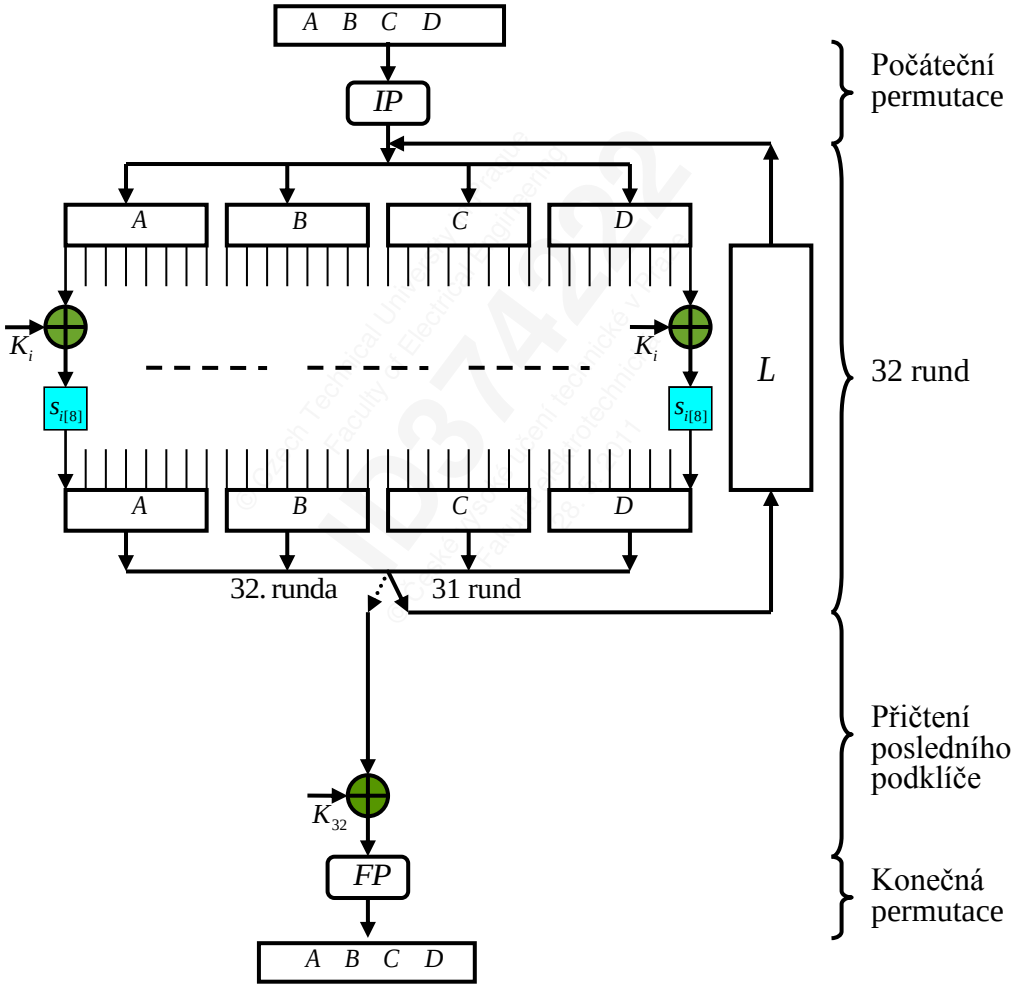
- velmi konzervativní návrh
- nepoužívá
  - datově závislé rotace, ani
  - násobení mod n, ani
  - sčítání mod n
- používá „tradiční“ operace
  - XOR
  - S-boxy
- vhodné pro čipové karty
- z počátečního klíče se spočítá 33 rundových klíčů
- pokud je klíč $k < 256$b, je doplněn jednou „1“ a více „0“ na celkovou délku 256b

Serpent - šifrování

- počáteční permutace (mění pořadí bitů v bloku)
- 32 rund, každá obsahuje :
  - xor s rundovým klíčem
  - průchod S-boxem
  - lineární transformace
- konečná permutace, která je inverzí k počáteční

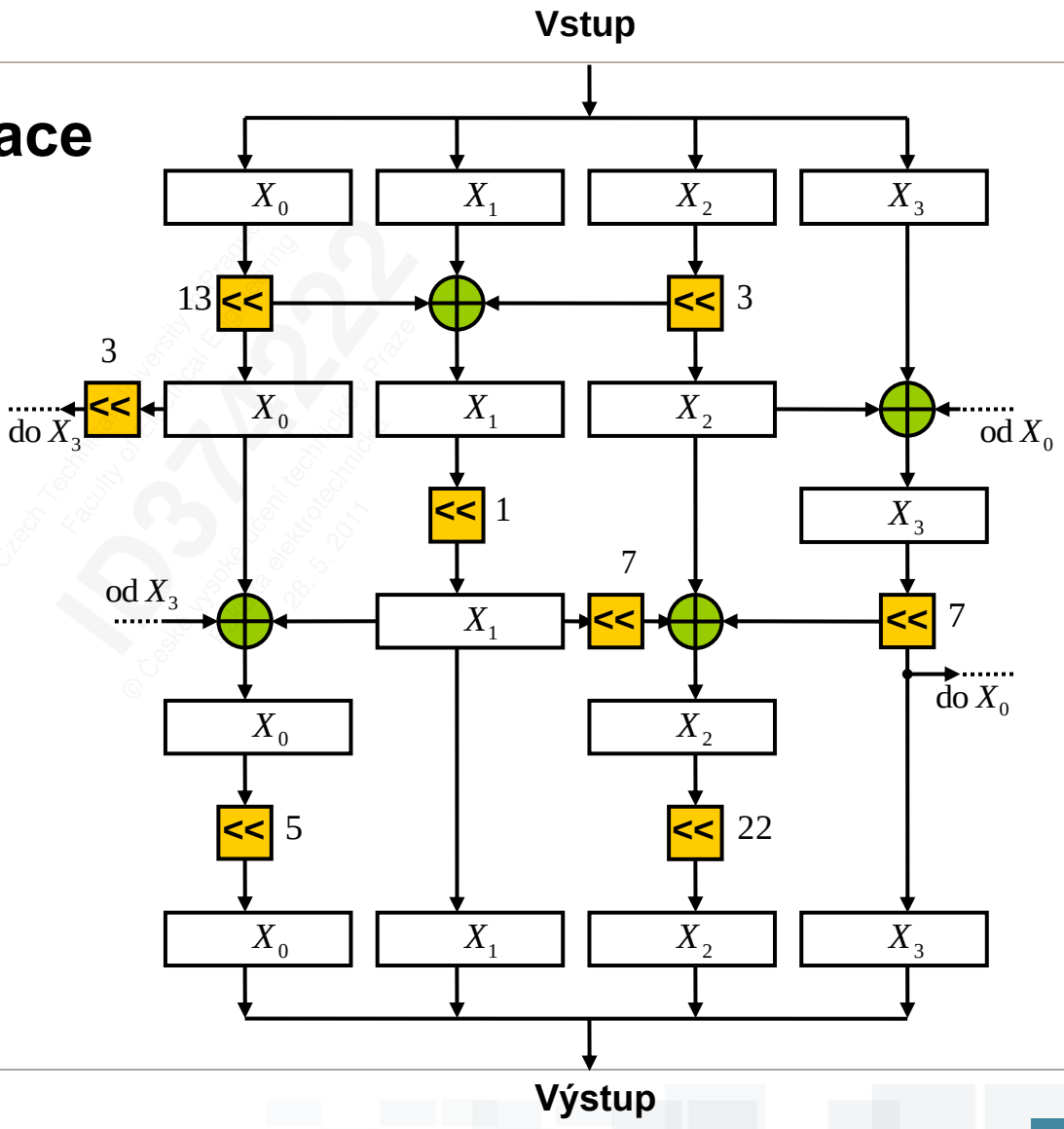
První a poslední krok nemají žádný význam z kryptografického hlediska. Slouží pouze k optimalizaci dat a zvyšují efektivitu výpočtů.

Serpent


A B C D
IP
A
B
C
D
K_i
K_i
$s_{i[8]}$
$s_{i[8]}$
L
A
B
C
D
32. runda
31 rund
K_{32}
FP
A B C D
Počáteční permutace
32 rund
Přičtení posledního podklíče
Konečná permutace

Serpent – blok L

Lineární transformace



Serpent

- Šifrování a dešifrování jsou dvě různé funkce, které nemají příliš mnoho společného.
- Implementace je proto přibližně dvakrát náročnější než implementace pouze jedné funkce.

Bezpečnost

- Již s 16 rundami je Serpent dostatečně odolný proti všem dnes známým útokům.
- Zvýšení počtu rund na 32 dále zvětšuje celkovou bezpečnost šifry.
- Pomocí diferenciální kryptoanalýzy a útoků se znalostí vybraných OT se povedlo prolomit Serpent s 6 rundami.

Twofish

Autoři: Bruce Schneier, John Kelsey, Doug Whiting
David Wagner, Chris Hall, Niels Ferguson

- 3. místo
- klasické Feistelovo schéma (jako DES)
- 16 rund
- klíč délky 128 až 256 bitů
- operace podobné jako v Rijndaelu
  - násobení v konečném poli F_2^8
  - sčítání mod 2^{32}
  - XOR
  - klíčově závislé S-boxy

Twofish

- vysoká roveň bezpečnosti, ale velmi složitý návrh
- podobně jako u Rijndaelu výkon klesá s délkou klíče

Operace v rundě

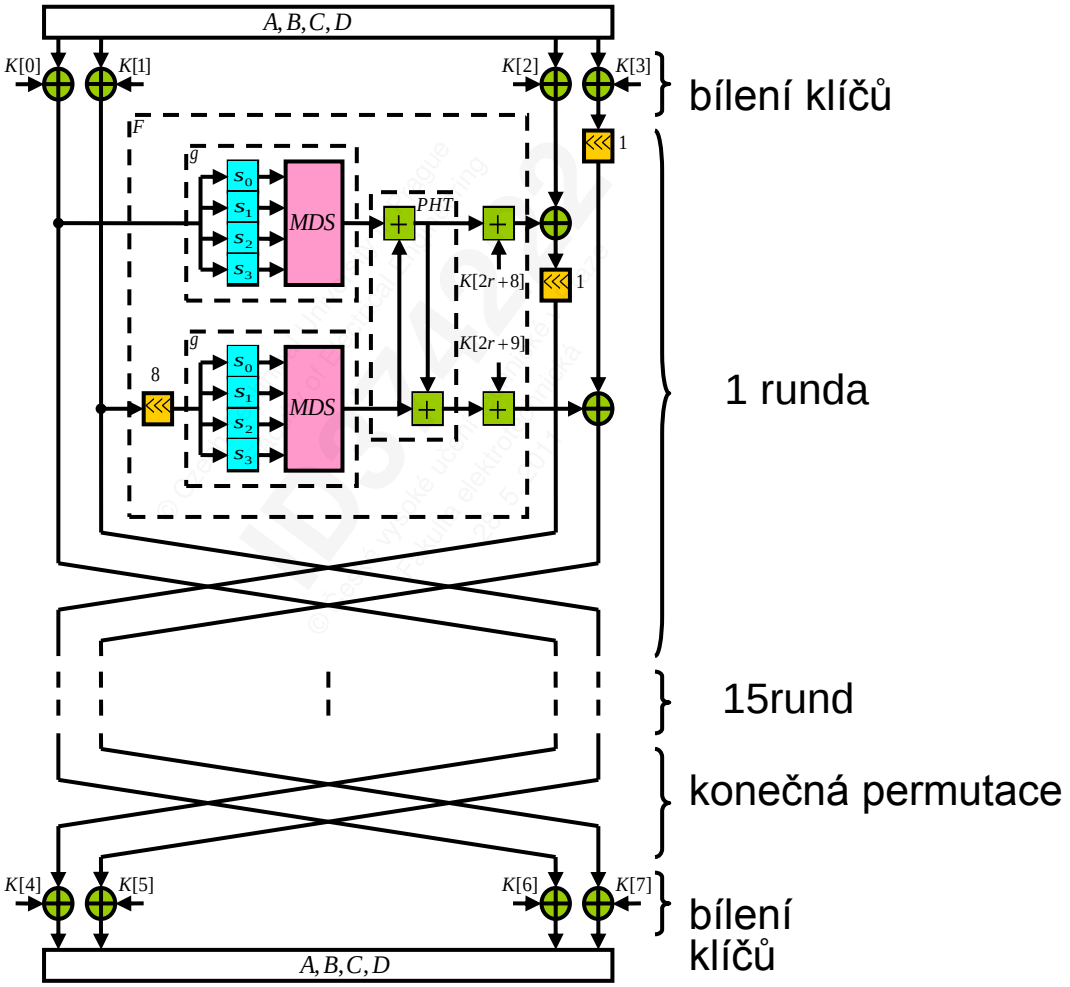
- 4 klíčově závislé S-boxy
- bitové rotace
- PHT - Pseudo-Hadamardova transformace
  - jednoduché míchání dvou vstupů, podle vzorce:
  - realizuje difúzi

$$a' = a + b \bmod 2^{32}$$

$$b' = a + 2b \bmod 2^{32}$$

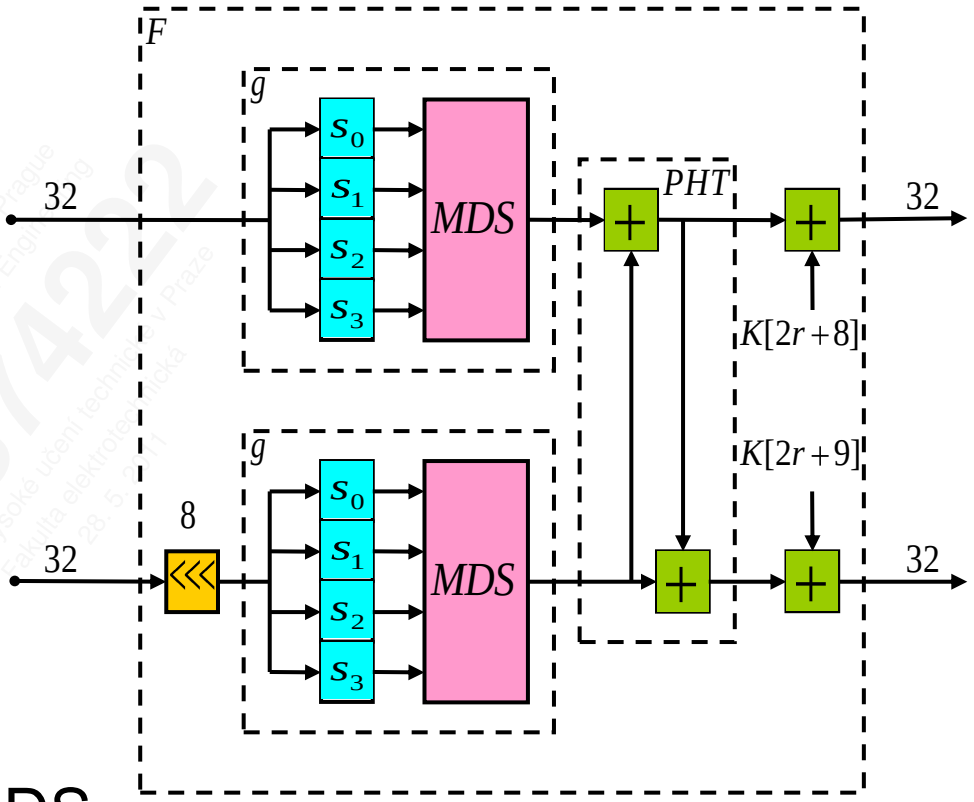
- polovina klíče je použita na šifrování a polovina modifikuje algoritmus (S-boxy)

Twofish


A, B, C, D
K[0]
K[1]
K[2]
K[3]
bílení klíčů
F
g
MDS
PHT
K[2r+8]
K[2r+9]
1
8
1 runda
15rund
konečná permutace
K[4]
K[5]
K[6]
K[7]
bílení klíčů
A, B, C, D

Twofish – operace v rundě

- z hlediska bezpečnosti je nejdůležitější funkce *g*
- vstupní 32bitové slovo je rozděleno na čtyři části
- každá čtvrtina vstupuje do jedné skupiny S-boxů
- každý S-box má 8 bitový vstup a výstup
- čtyři výstupy z S-boxů jsou chápány jako vektor v F_2^8 , který je vynásoben maticí MDS
- následuje přičtení rundových klíčů
- výsledek je interpretován jako 32bitové slovo



Twofish – operace v rundě

- **MDS** - Maximum Distance Separable
- matice 4x4 bajtů, kterou násobíme 32 bitový vstup

$$\begin{bmatrix} 01 & EF & 5B & 5B \\ 5B & EF & EF & 01 \\ EF & 5B & 01 & EF \\ EF & 01 & EF & 5b \end{bmatrix}$$

- matice je MDS, pokud realizuje lineární transformaci ( ) a ( ) taková, že žádné dvě rozdílné -tice ( ) se neliší více než v prvcích (– konečné pole).

- dešifrování je téměř identické s šifrováním
- velmi složitá příprava klíčů

Twofish – bezpečnost

- neexistuje žádný útok na plnou verzi Blowfish
- 1999 – Niels Ferguson – prakticky nerealizovatelný útok na Twofish se 6 rundami (z 16) a 256bitovým klíčem vyžadující 2^{256} kroků
- 2010 – teoretický útok zobecněnou variantou diferenciální kryptoanalýzy (truncated differential analysis) na plnou verzi Twofish
  - PRST odhadu správné diference tímto útokem je asi $2^{-57,3}$ na blok
  - na nalezení dobré dvojice diferencí je potřeba vygenerovat zhruba 2^{51} vybraných OT (~32PB dat)
- lze prolomit Blowfish s 10 rundami útok pomocí vybraných klíčů (bez bělení klíčů)

Rijndael (AES)

Autoři: Vincent **Rij**men a Joan **Dae**men

- Belgie
- iterovaná bloková šifra (stejně jako DES)
- není šifra Feistelova typu (na rozdíl od DESu)
- substitučně-permutační síť
- veškeré matematické operace v AESu se odehrávají v konečném poli $F_2{}^8$ s nerozložitelným polynomem

$$F(x) = x^8 + x^4 + x^3 + x + 1$$

Příklad: $\{53\} \bullet \{CA\} = \{01\}$ v poli $F_2{}^8$, protože

$(x^6 + x^4 + x + 1)(x^7 + x^6 + x^3 + x) =$

$x^{13} + x^{12} + x^9 + x^7 + x^{11} + x^{10} + x^7 + x^5 + x^8 + x^7 + x^4 + x^2 + x^7 + x^6 + x^3 + x =$

$x^{13} + x^{12} + x^{11} + x^{10} + x^9 + x^8 + x^6 + x^5 + x^4 + x^3 + x^2 + x$

a $x^{13}+x^{12}+x^{11}+x^{10}+x^9+x^8+x^6+x^5+x^4+x^3+x^2+x \bmod x^8+x^4+x^3+x+1 = 1$

Protože výsledkem násobení $\{53\}$ a $\{CA\}$ je 1,
$\{53\}$ je multiplikativní inverze $\{CA\}$.

Rijndael (AES) v číslech

- **Délka bloku:** **128***, 192 nebo 256 bitů
- **Délka klíče:** **128***, **192** nebo **256** bitů
  - nezávisle na délce bloku
- 10 až 14 rund (v závislosti na délce klíče a bloku)
- v každé rundě se provádějí čtyři operace:
  - SubByte (nelineární operace)
  - ShiftRow (lineární operace)
  - MixColumn (nelineární operace)
  - AddRoundKey (lineární operace)

Nr	N_b=4	N_b=6	N_b=8
N_k=4	10	12	14
N_k=6	12	12	14
N_k=8	14	14	14

***v AES standardu**

AES v číslech

- všechny operace v AES se provádějí na 2-D poli označovaném jako Stav (State)
- pole má vždy 4 řádky a 4 reps.6 resp. 8 sloupců
- počet sloupců závisí na velikosti bloku
- každá buňka pole obsahuje 1 byte dat
- celková velikost stavu je 128/192/256 bitů

$S[r,c]=in[r + 4c]$ pro $0 \le r < 4$ a $0 \le c < N_b$ N_b = délka bloku/32

Vstupní byty

In_0	In_4	In_8	In_{12}
In_1	In_5	In_9	In_{13}
In_2	In_6	In_{10}	In_{14}
In_3	In_7	In_{11}	In_{15}

Pole stavů

$S_{0,0}$	$S_{0,1}$	$S_{0,2}$	$S_{0,3}$
$S_{1,0}$	$S_{1,1}$	$S_{1,2}$	$S_{1,3}$
$S_{2,0}$	$S_{2,1}$	$S_{2,2}$	$S_{2,3}$
$S_{3,0}$	$S_{3,1}$	$S_{3,2}$	$S_{3,3}$

Výstupní byty

Out_0	Out_4	Out_8	Out_{12}
Out_1	Out_5	Out_9	Out_{13}
Out_2	Out_6	Out_{10}	Out_{14}
Out_3	Out_7	Out_{11}	Out_{15}

Rijndael (AES)

Volitelná délka klíče

- 128 bitů … $3{,}4 \cdot 10^{18}$ klíčů
- 192 bitů … $6{,}2 \cdot 10^{57}$ klíčů
- 256 bitů … $1{,}1 \cdot 10^{77}$ klíčů

Volitelná délka bloku

- 128 bitů
- 192 bitů
- 256 bitů

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$	$k_{0,4}$	$k_{0,5}$	$k_{0,6}$	$k_{0,7}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$	$k_{1,4}$	$k_{1,5}$	$k_{1,6}$	$k_{1,7}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$	$k_{2,4}$	$k_{2,5}$	$k_{2,6}$	$k_{2,7}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$	$k_{3,4}$	$k_{3,5}$	$k_{3,6}$	$k_{3,7}$

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$	$a_{0,4}$	$a_{0,5}$	$a_{0,6}$	$a_{0,7}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$	$a_{1,4}$	$a_{1,5}$	$a_{1,6}$	$a_{1,7}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$	$a_{2,4}$	$a_{2,5}$	$a_{2,6}$	$a_{2,7}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$	$a_{3,4}$	$a_{3,5}$	$a_{3,6}$	$a_{3,7}$

Rijndael (AES) - algoritmus

Průběh operací v Rijndaelu

```
Rijndael(Stav, Klíč) {
    ExpanzeKlíče(Klíč, ExpandovanýKlíč);
    AddRoundKey(Stav, ExpandovanýKlíč);
    for (i=1; i<10; i++)
        Runda(Stav, ExpandovanýKlíč +4);
    PosledníRunda(State, ExpandovanýKlíč
    +4x10);
}
```

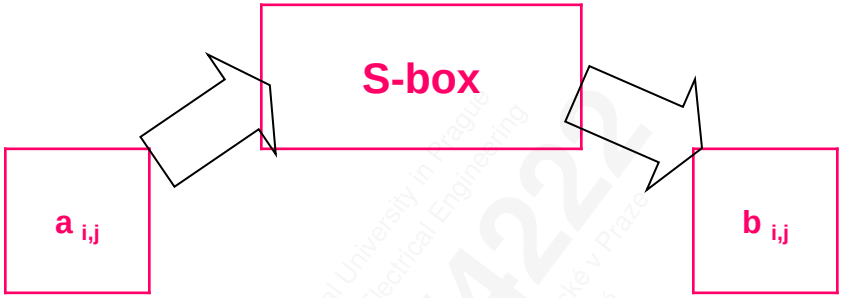
Operace v rundě

```
Runda(Stav, RundovýKlíč) {
    ByteSub(Stav);
    ShiftRow(Stav);
    MixColumn(Stav);
    AddRoundKey(Stav,
    RundovýKlíč);
}
```

Stav	- pole 4 slov (každé 32 bitů)
Počet_Rund	- pro variantu blok/klíč =128 bitů má hodnotu 10
ExpanzeKlíče	- algoritmus který z 128 bitů klíče vyrobí 1408 bitů (obsahuje operace XOR, vyhledávání v S-boxu, a rotaci v rámci slova)
AddRoundKey	- XOR klíče a OT – tzv. bílení (whitening)
PosledníRunda	- stejná jako ostatní rundy, ale neobsahuje MixColumn

Rijndael (AES) – ByteSub

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$



$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

Cíl: vnesení nelinearity do procesu šifrování

ByteSub je nelineární operace ve dvou krocích:

1) na každý byte se v F_2^8 aplikuje multiplikativní inverze

2) na každý byte se aplikuje affiní transformace (nad F_2) ve tvaru

$$8F \bullet a_{i,j} \oplus A6$$

- operace SubByte má v AESu stejný význam jako „S-box“ v DESu
- může být implementován jako tabulka pro každý byte

Rijndael (AES) – ByteSub

Pohled na ByteSub jako na S-box.

a_{ij}^{-1}

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

S-box

Rijndael (AES) – ByteSub

Tabulka pro operaci ByteSub.

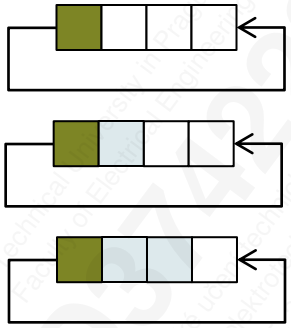
příklad: S_{in}={7b}

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Rijndael (AES) – ShiftRow

$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,0}$	$b_{1,1}$	$b_{1,2}$	$b_{1,3}$
$b_{2,0}$	$b_{2,1}$	$b_{2,2}$	$b_{2,3}$
$b_{3,0}$	$b_{3,1}$	$b_{3,2}$	$b_{3,3}$

žádná rotace



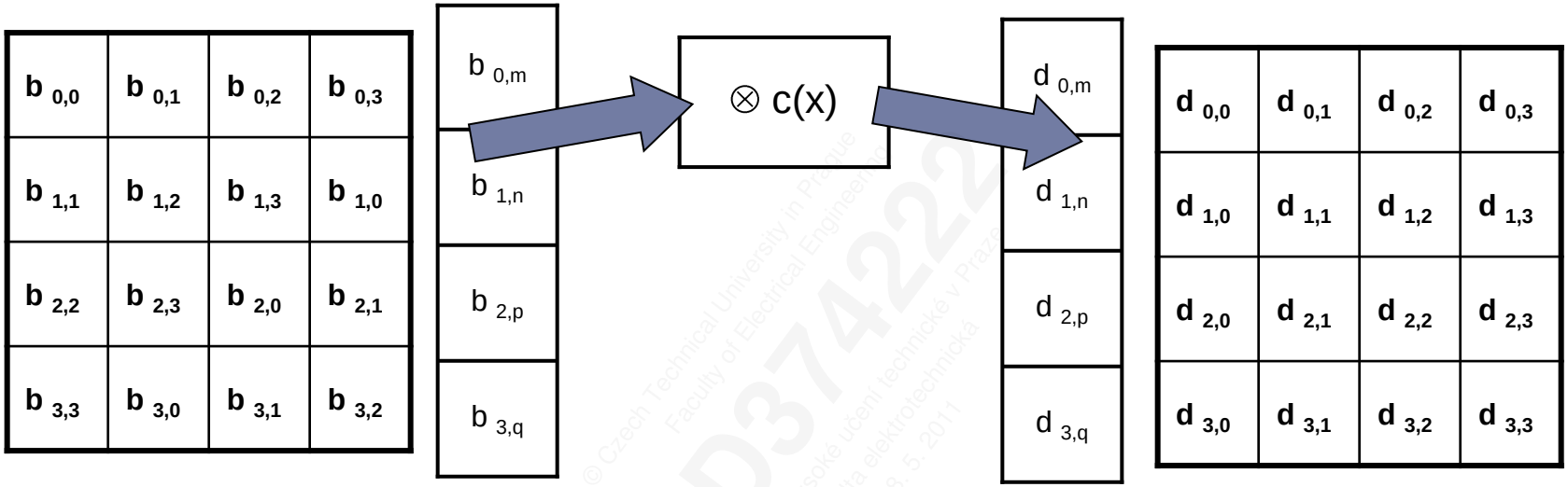
$b_{0,0}$	$b_{0,1}$	$b_{0,2}$	$b_{0,3}$
$b_{1,1}$	$b_{1,2}$	$b_{1,3}$	$b_{1,0}$
$b_{2,2}$	$b_{2,3}$	$b_{2,0}$	$b_{2,1}$
$b_{3,3}$	$b_{3,0}$	$b_{3,1}$	$b_{3,2}$

Operace ShiftRow se provádí na jednotlivých řádcích.

Cíl: difúze mezi sloupci

	C_0	C_1	C_2	C_3
$N_b = 4$	0	1	2	3
$N_b = 6$	0	1	2	3
$N_b = 8$	0	1	3	4

AES – MixColumn



- operace MixColumn pracuje se sloupci
- každý sloupec se uvažuje jako polynom nad F_2^8 a je vynásoben s polynomem $c(x) = \{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\}$ mod x^4+1
- Implementuje se pomocí XOR

Cíl: zajištění difůze mezi jednotlivými byty.
Společně s ShiftRow zajistí tzv. lavinovitý efekt
Koeficienty matice byly zvoleny také s ohledem na možnost efektivní implementace.

$$c(x) = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$$

AES – AddRoundKey

$d_{0,0}$	$d_{0,1}$	$d_{0,2}$	$d_{0,3}$
$d_{1,0}$	$d_{1,1}$	$d_{1,2}$	$d_{1,3}$
$d_{2,0}$	$d_{2,1}$	$d_{2,2}$	$d_{2,3}$
$d_{3,0}$	$d_{3,1}$	$d_{3,2}$	$d_{3,3}$

$\oplus$

$k_{0,0}$	$k_{0,1}$	$k_{0,2}$	$k_{0,3}$
$k_{1,0}$	$k_{1,1}$	$k_{1,2}$	$k_{1,3}$
$k_{2,0}$	$k_{2,1}$	$k_{2,2}$	$k_{2,3}$
$k_{3,0}$	$k_{3,1}$	$k_{3,2}$	$k_{3,3}$

=

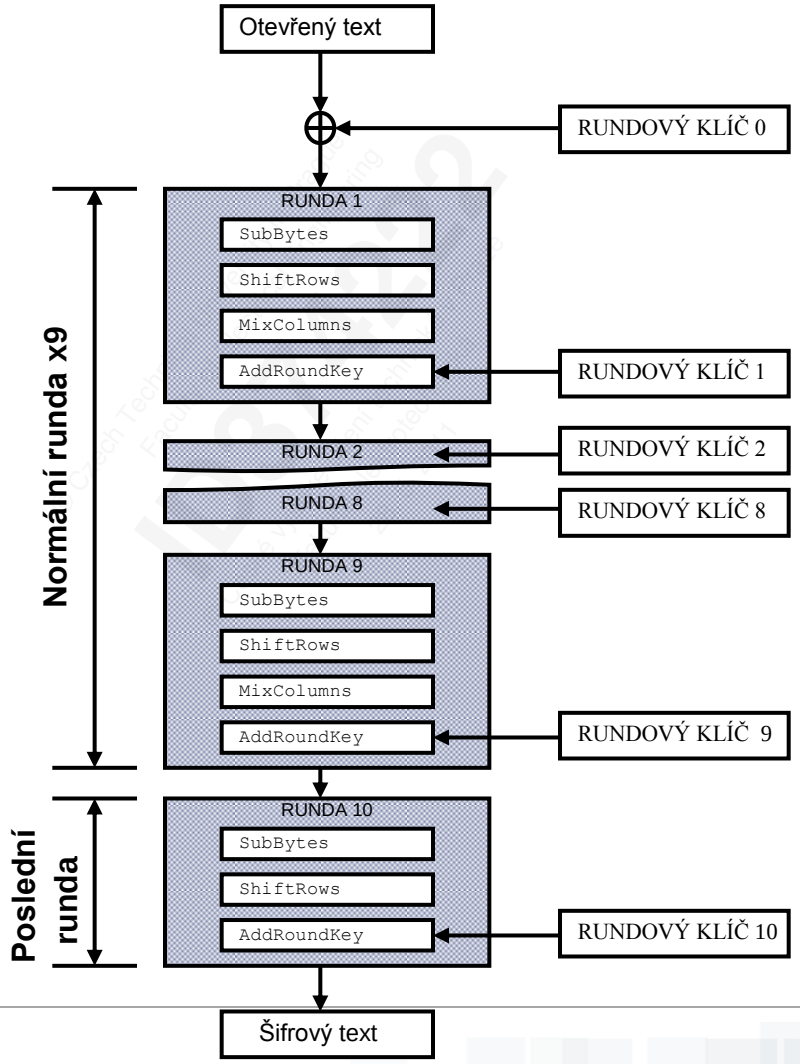
$e_{0,0}$	$e_{0,1}$	$e_{0,2}$	$e_{0,3}$
$e_{1,0}$	$e_{1,1}$	$e_{1,2}$	$e_{1,3}$
$e_{2,0}$	$e_{2,1}$	$e_{2,2}$	$e_{2,3}$
$e_{3,0}$	$e_{3,1}$	$e_{3,2}$	$e_{3,3}$

Operace AddRoundKey provádí přičtení rundového klíče ke Stavu.
Klíč rundy je určen pomocí plánovacího algoritmu (key schedule).

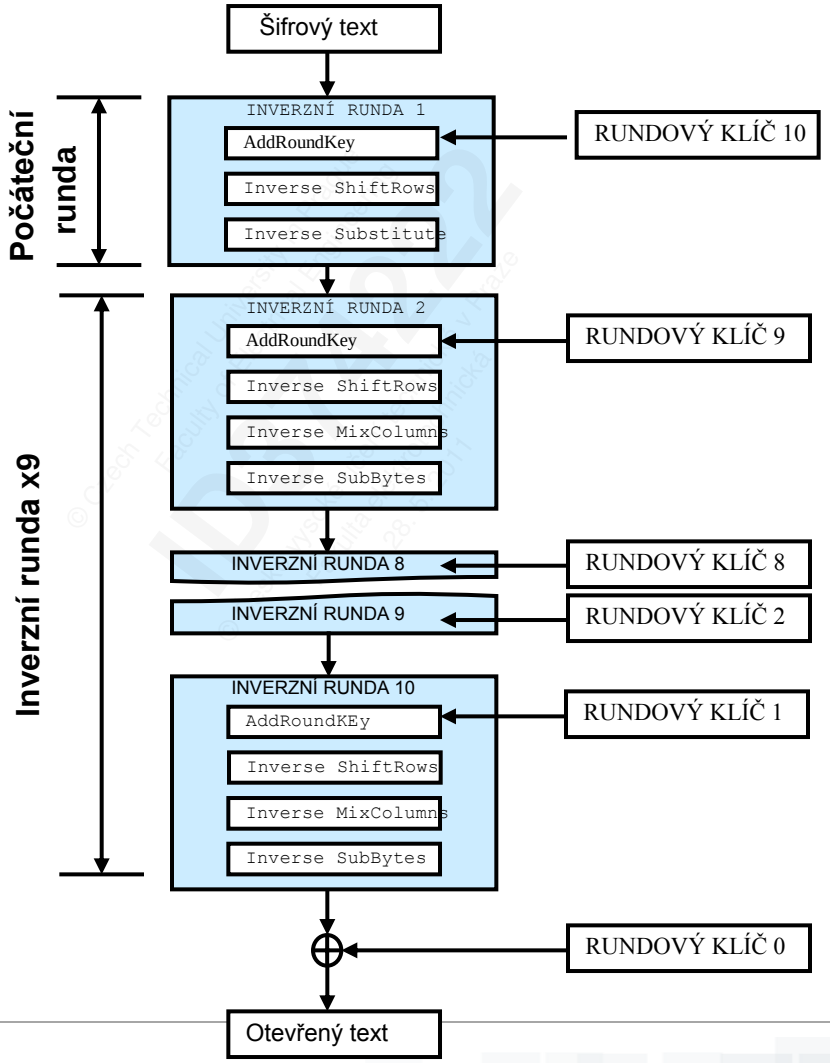
Cíl: operace v rundě musí být klíčově závislé.

XOR OT (nebo ŠT) a klíče se nazývá bílení (whitening) klíče. Je to jednoduchý postup zvyšující bezpečnost. Brání útočníkovi vytvářet odpovídající páry OT-ŠT. U Rijndaelu je realizován před první rundou.

Rijndael (AES) – šifrování

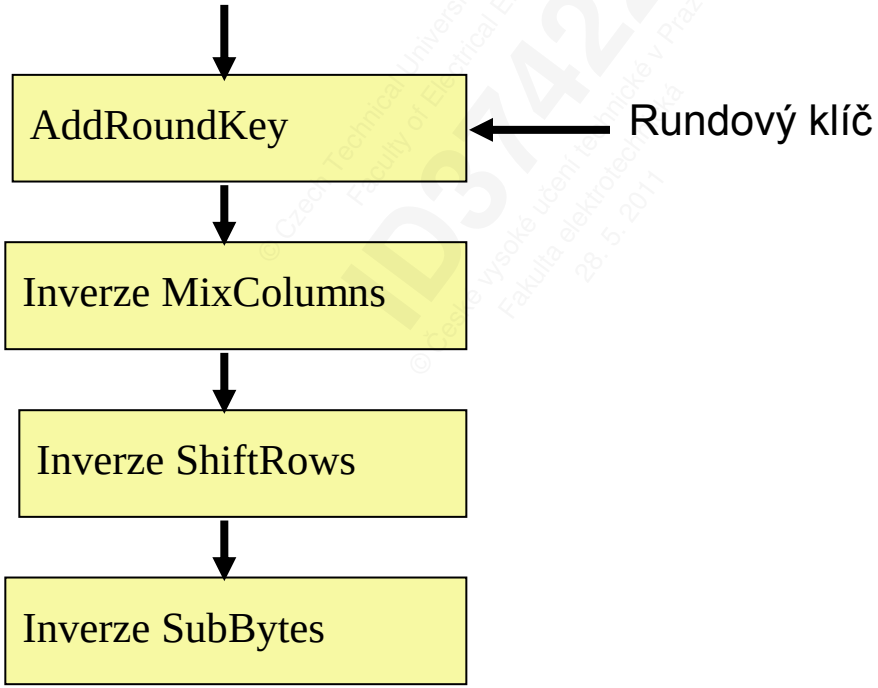


Rijndael (AES) – Dešifrování



Rijndael (AES) – dešifrování

Při dešifrování probíhají inverze jednotlivých operací v opačném pořadí než při šifrování:



Rijndael (AES) – Dešifrování

- inverzní operace se provádějí v opačném pořadí než při šifrování
- kromě nelineární operace SubBytes je inverze zbylých operací velmi jednoduchá
- Rijndael je navržen tak, že lze použít stejný kód na šifrování i dešifrování
- pouze se zamění příslušné tabulky a polynomy (v každém ze 4 kroků)
- zbytek operací probíhá jako při šifrování

AES – dešifrování jednotlivých operací

- Operace AddRoundKey je invertibilní
  - operace $\oplus$ je inverzní sama k sobě tzn. po provedení opětovného přičtení polynomu mod 2 dostaneme původní polynom
- MixColumn je invertibilní
  - inverzi se realizuje pomocí násobení inverzním polynomem

$$c(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}$$

- ShiftRow je invertibilní
  - Inverze se realizuje jako cyklický posun doleva
- Nelineární operace ByteSub je také vratná
  - inverze je implementována pomocí vyhledávání v tabulce

AES – dešifrování

Inverze operace ByteSub

- Příklad S_{in}={21}

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

AES - optimalizace

- Určeno pro systémy pracující s 32 bitovým vstupem (nebo větším).
- Urychlení algoritmu zkombinováním operací SubBytes a ShiftRows spolu s MixColumns a jejich transformací do několika vyhledání v tabulkách.
- Je nutné sestavit čtyři tabulky s rozměry 16x16, což zabere celkem 4x16x16x8=4096 bytů paměti.
- Rundu pak můžeme realizovat pomocí 16 vyhledání v tabulce a 12 32bitových operací XOR
- Poté následují čtyři 32bitové operace XOR s klíčem (operace AddRoundKey)

AES – bezpečnost

- v roce 2002 byl představen teoretický útok proti AESu
  XSL (eXtended Sparse Linearization)
  - Nicolas Courtois a Josef Pieprzyk (spoluautor algoritmů Loki, Loki97, HAVAL)
  - vyjádření algoritmu jako soustavy rovnic – pro AES-128 jde o přibližně 8000 rovnic o 1600 neznámých
  - obecně se jedná o NP problém označovaný jako MQE (multivariate quadratic equations)
  - v roce 2005 byl podán důkaz (Cid, Leurent), že v současné podobě není XSL efektivní nástroj pro útok na AES (i kdyby fungoval, jeho režie je tak velká, že výsledek není lepší než útok hrubou silou)
  - Několik dalších kryptologů také poukázalo na problémy v matematickém pozadí útoku XSL a prezentované závěry autorů útoku označili za chybné.
  - V každém případě popsaný útok nelze v praxi zrealizovat.
- do roku 2006 nebyl znám žádný praktický útok na AES
- od roku 2006 známá možnost útoků proti implementacím AES tzv. postraními kanály
  - útok postraním kanálem neútočí na samotný algoritmus, ale jeho implementaci

AES – bezpečnost

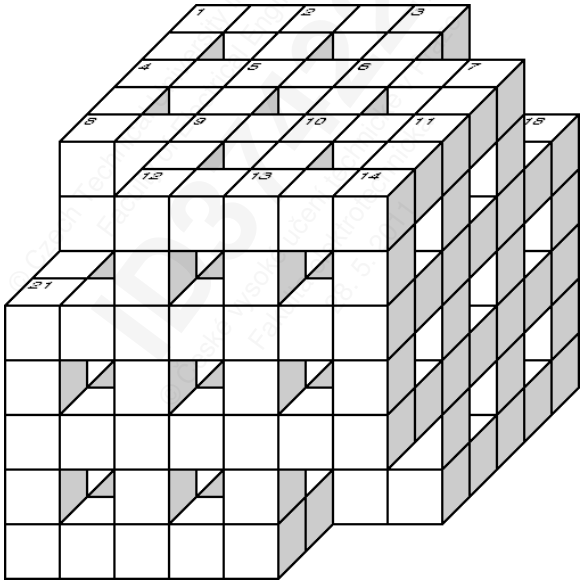
AES-128

- není znám žádný útok s časovou složitostí menší než 2^{128}

AES-192, AES-256

- popsány útoky s časovou složitostí 2^{172} a 2^{119} kroků
- funguje pouze se speciálním typem útoků
  - „related key“ útok
  - kryptoanalytik musí mít k dispozici OT zašifrovaném mnoha klíči, mezi kterými je vhodná vazba
- týká se jen AES-256 s 10 rundami a AES-192 s 9 rundami
- princip TMTO (Time Memory TradeOff)
- úspora v času znamená zvýšené nároky na pamět
- prostorová složitost útoku na AES-256 je 2^{119}
- nelze prakticky zrealizovat

Srovnání kandidátů na AES



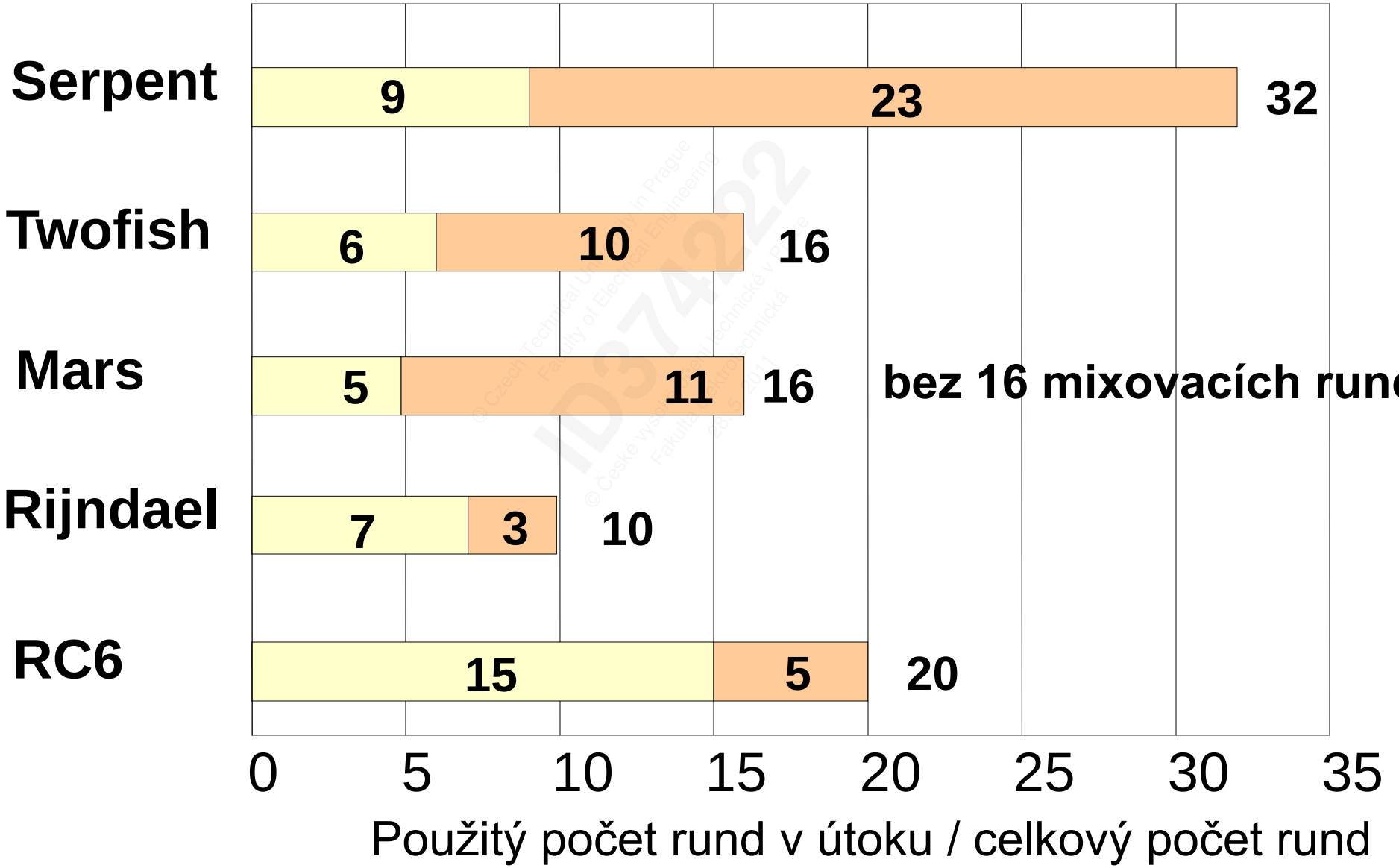
	XOR	Table / S-box	FixS/R	VarS/R	Add	Sub	Mul	GFmul
Mars	*	*	*	*	*	*	*	
RC6	*		*	*	*		*	
Rijndael	*	*	*					*
Serpent	*	*	*					
Twofish	*	*	*		*			*

Legenda:	
XOR	operece exkluzivní OR
Table / S-box	vyhledávání v tabulce nebo v S-boxu
FixS/R	Pevný posun nebo rotace
VarS/R	Posun nebo rotace o proměnný počet bitů
Add	Sčítání mod 2^{32}
Sub	Odčítání mod 2^{32}
Mul	Násobení mod 2^{32}
GFmul	Násobení v konečném poli F_2^8
*	Operace používané v daném algoritmu

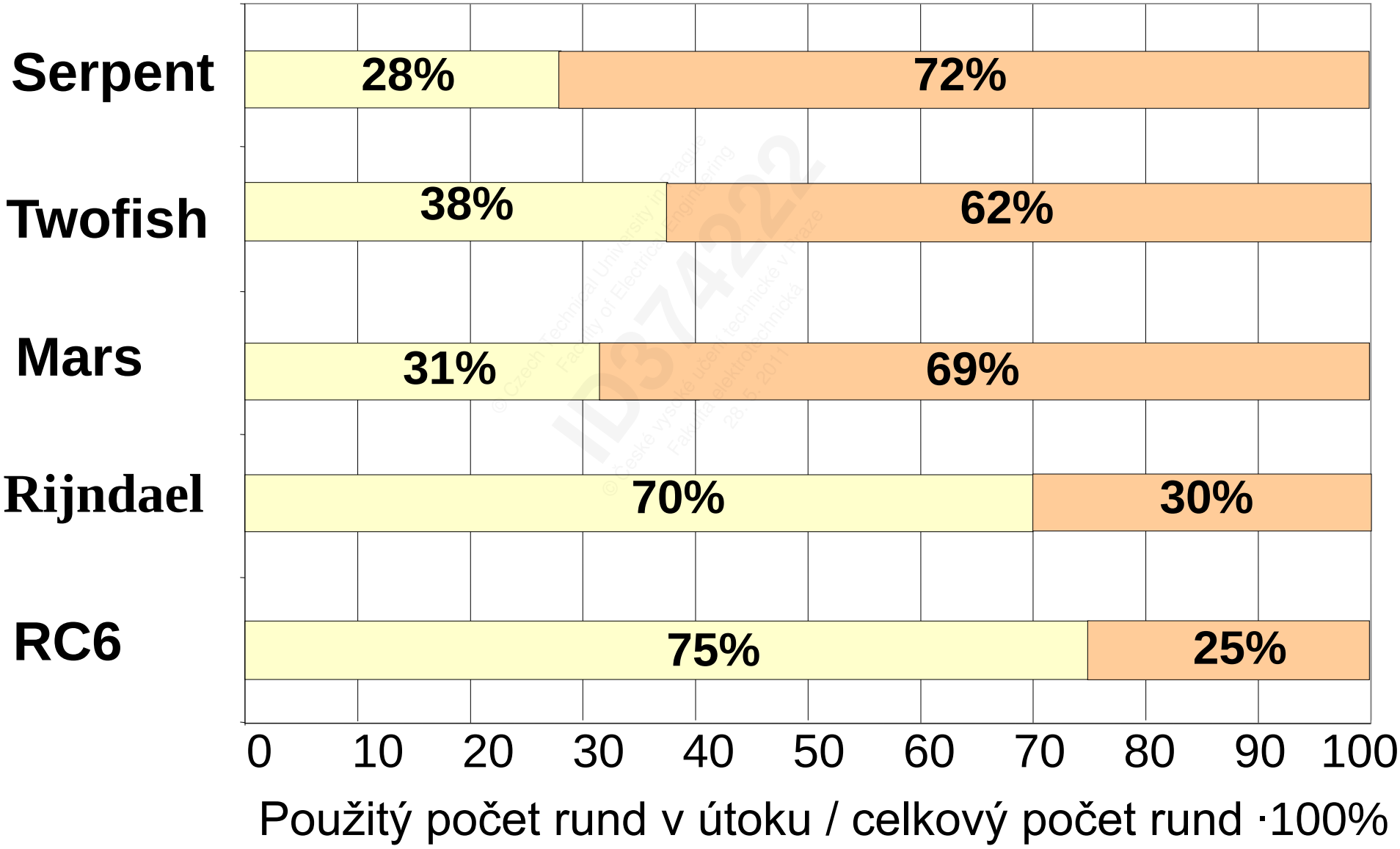
Útoky na finalisty

Algoritmus, Počet rund	Počet rund [délka klíče]	Typ útoku	Počet OT	Operací
Mars 16 Core (C) 16 Mixovacích (M)	11C	Amp. Boomerang	2^{65}	2^{229}
	16M,5C	Meet-in-Middle	8	2^{232}
RC6 20	14	Statistic	2^{118}	2^{112}
	15 [256]	Statistic	2^{119}	2^{215}
Rijndael 10 (128) 12 (192) 14 (256)	7 [192, 256]	Differential	2^{32}	2^{140}
	8 [256]	Differential	$2^{128} - 2^{119}$	2^{204}
	9 [256]	Related key	2^{77}	2^{224}
Serpent 32	8 [192, 256]	Boomerang	2^{128}	2^{163}
	9 [192, 256]	Amp. boomerang	2^{110}	2^{252}
Twofish 16	5 no post-whit	Related key	$2^{22.5}$	2^{51}
	6 [256]	Differential	NA	2^{256}

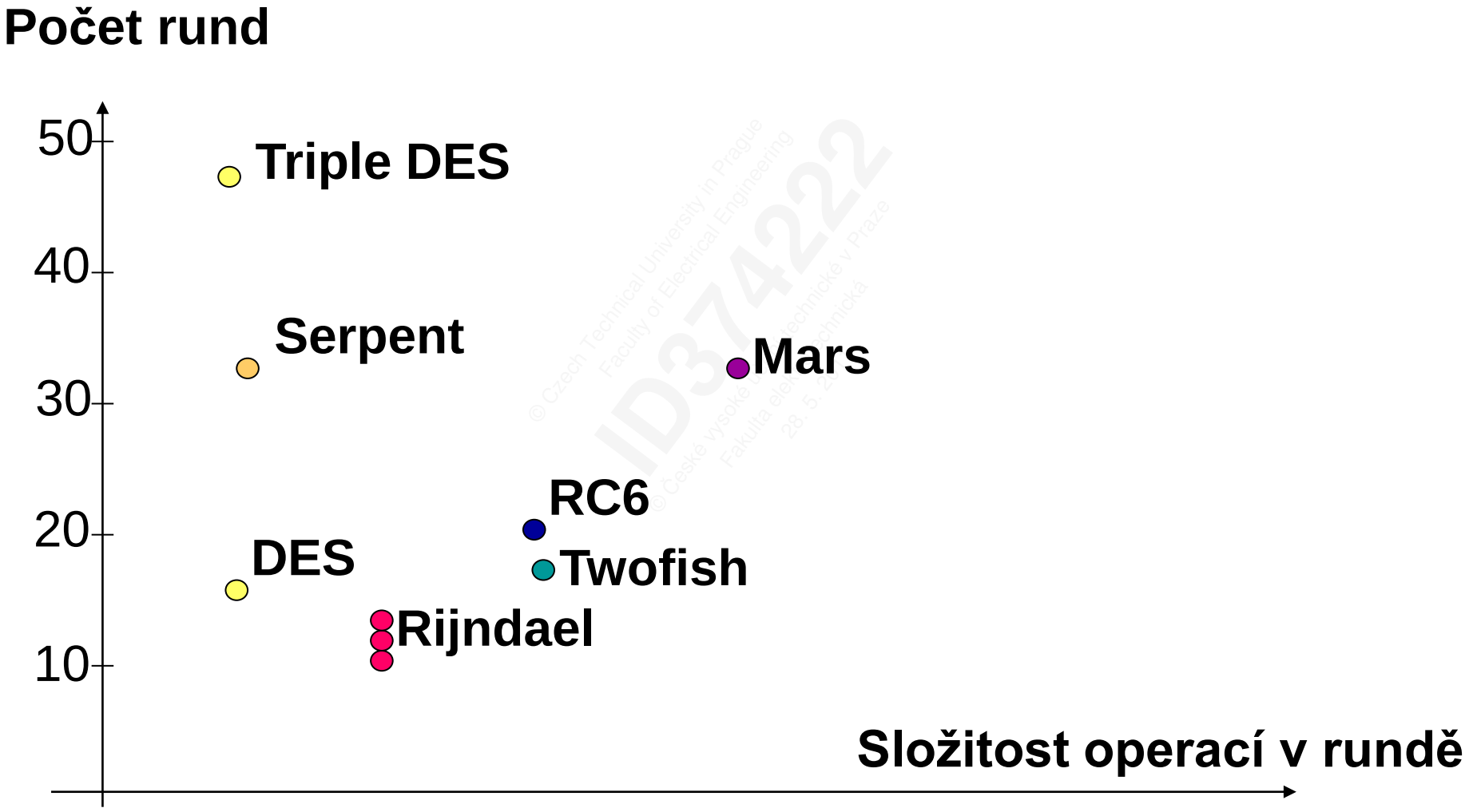
Bezpečnost: brute force nebo nějaký chytrý byť teoretický útok

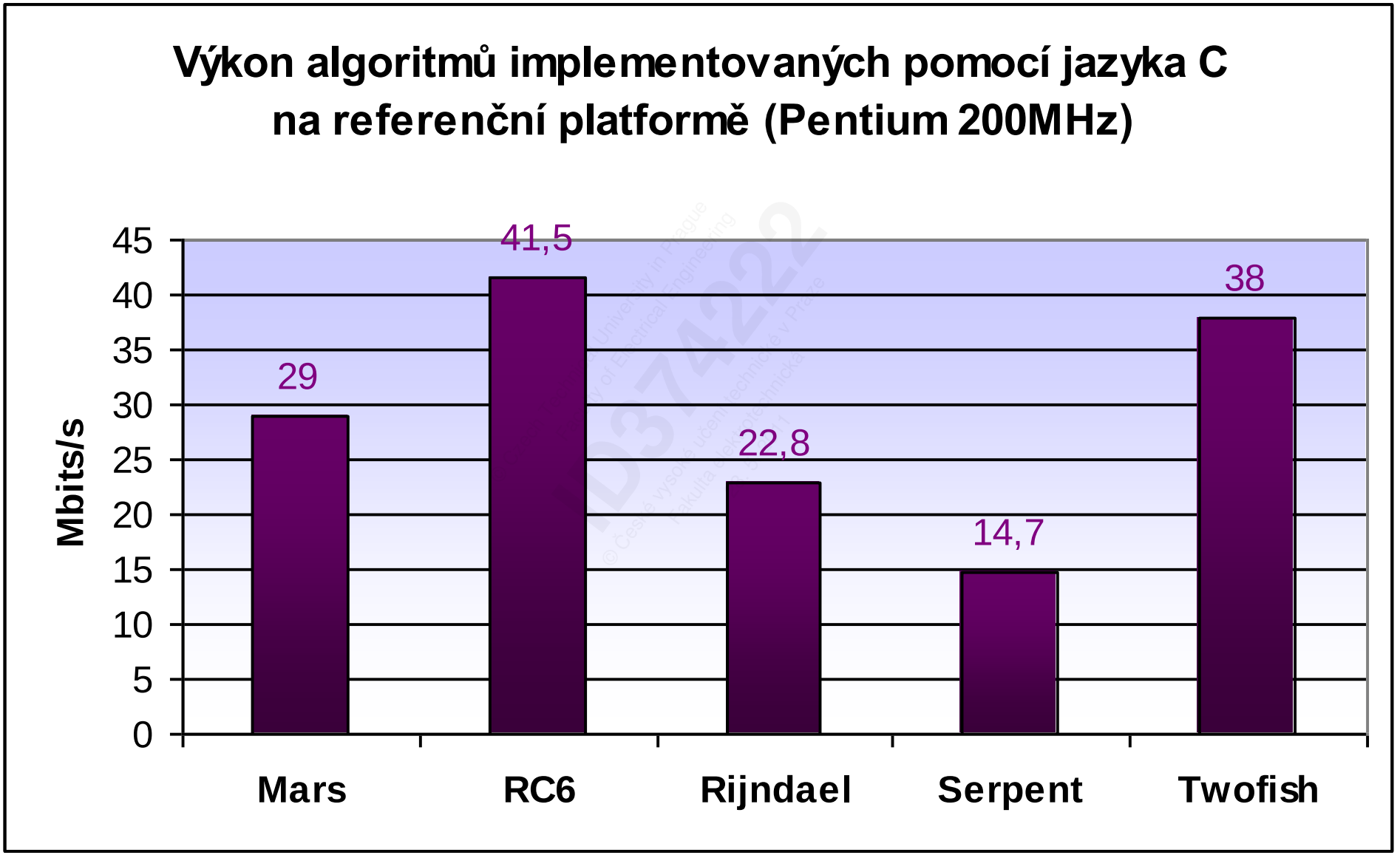


Bezpečnost: brute force nebo nějaký chytrý byť teoretický útok



Počet rund vs. složitost rundy




Výkon algoritmů implementovaných pomocí jazyka C
na referenční platformě (Pentium 200MHz)
Mbits/s
45
40
35
30
25
20
15
10
5
0
29
41,5
22,8
14,7
38
Mars
RC6
Rijndael
Serpent
Twofish

Srovnání AES a ostatních finalistů (DP)

Co a jak se testovalo (a na čem):

Režim činosti : ECB, CBC

Funkce: Šifrování, dešifrování, generování klíčů

Kompilátor/OS/CPU: GCC/Linux/Pentium 3
Dev-C++/Windows XP/AMD Athlon

Délka klíče: 128, 192, 256 bitů

Velikost vstupních dat: malý vstup (1 blok, 16 bloků),
velký vstup (8192 a 32768 bloků)

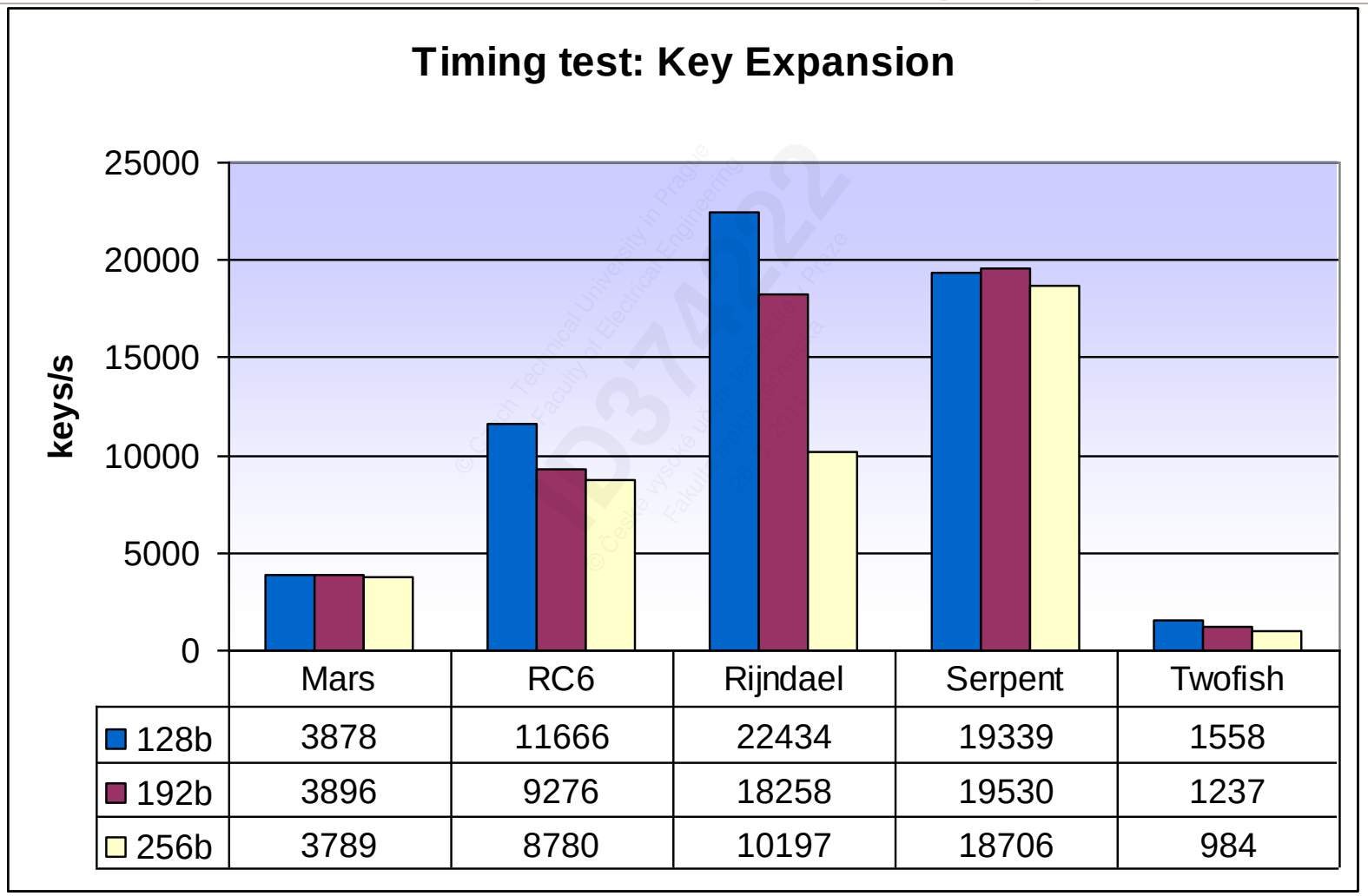
Co se měřilo:

- počet cyklů CPU
- čas

Pro korektní srovnání jednotlivých algoritmů bylo nutné omezit vliv ostatních částí OS, cache CPU , které mohly ovlivnit výsledek.

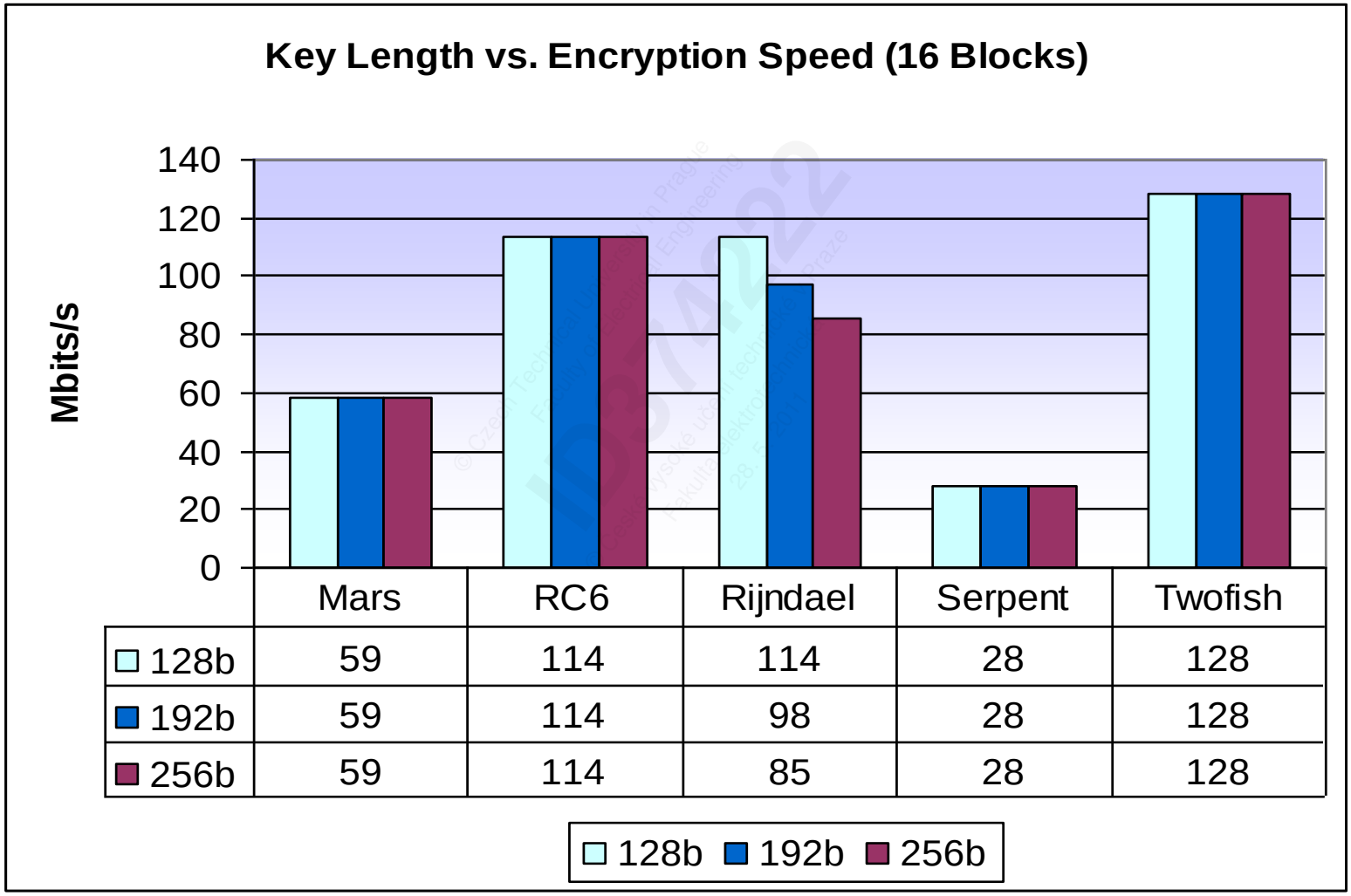
- 200 testů pro každou kombinaci
- uvažovaly se pouze výsledky medián σ

Srovnání AES a ostatních finalistů (DP)



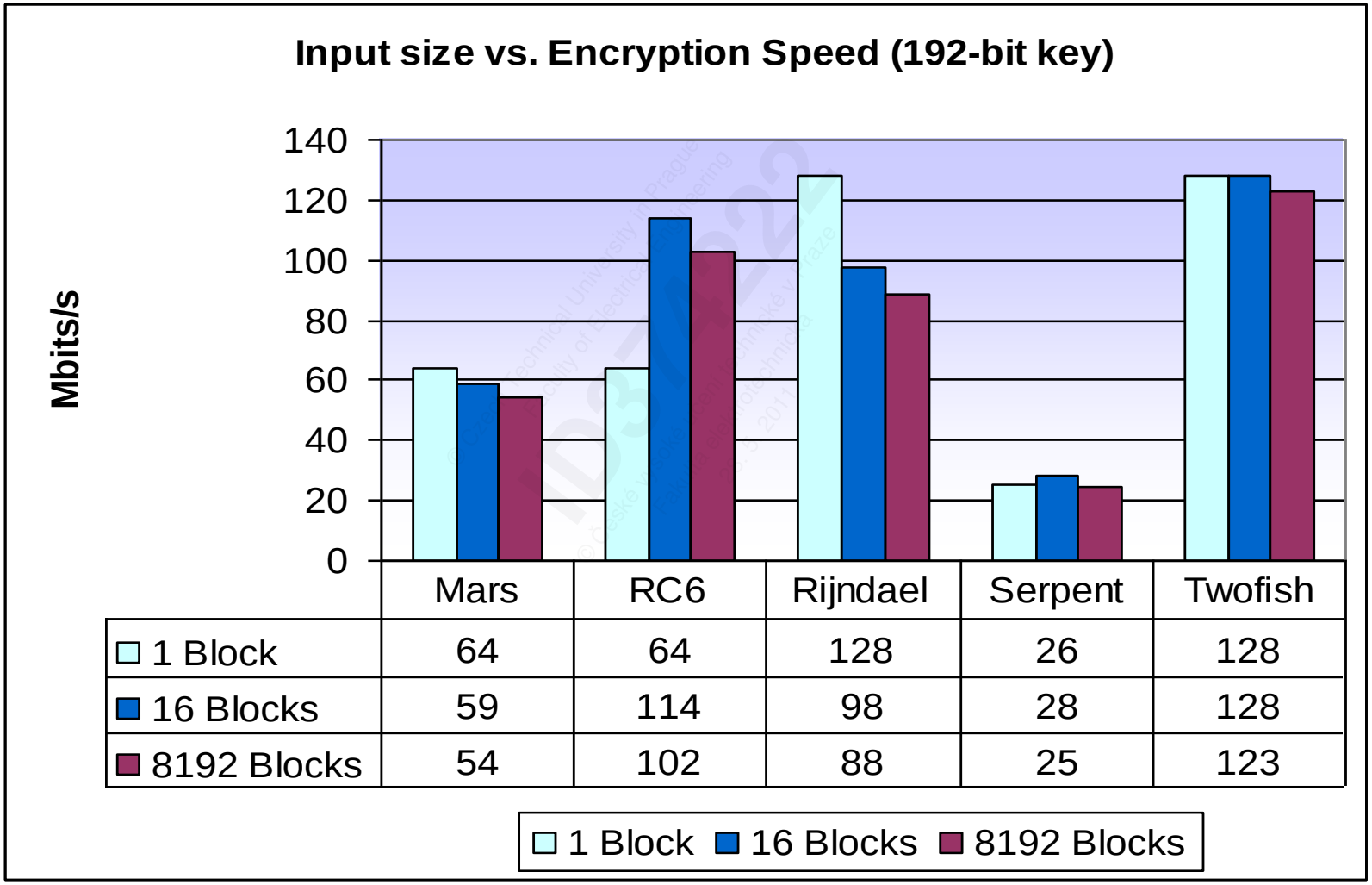
	Mars	RC6	Rijndael	Serpent	Twofish
128b	3878	11666	22434	19339	1558
192b	3896	9276	18258	19530	1237
256b	3789	8780	10197	18706	984

Srovnání AES a ostatních finalistů (DP)



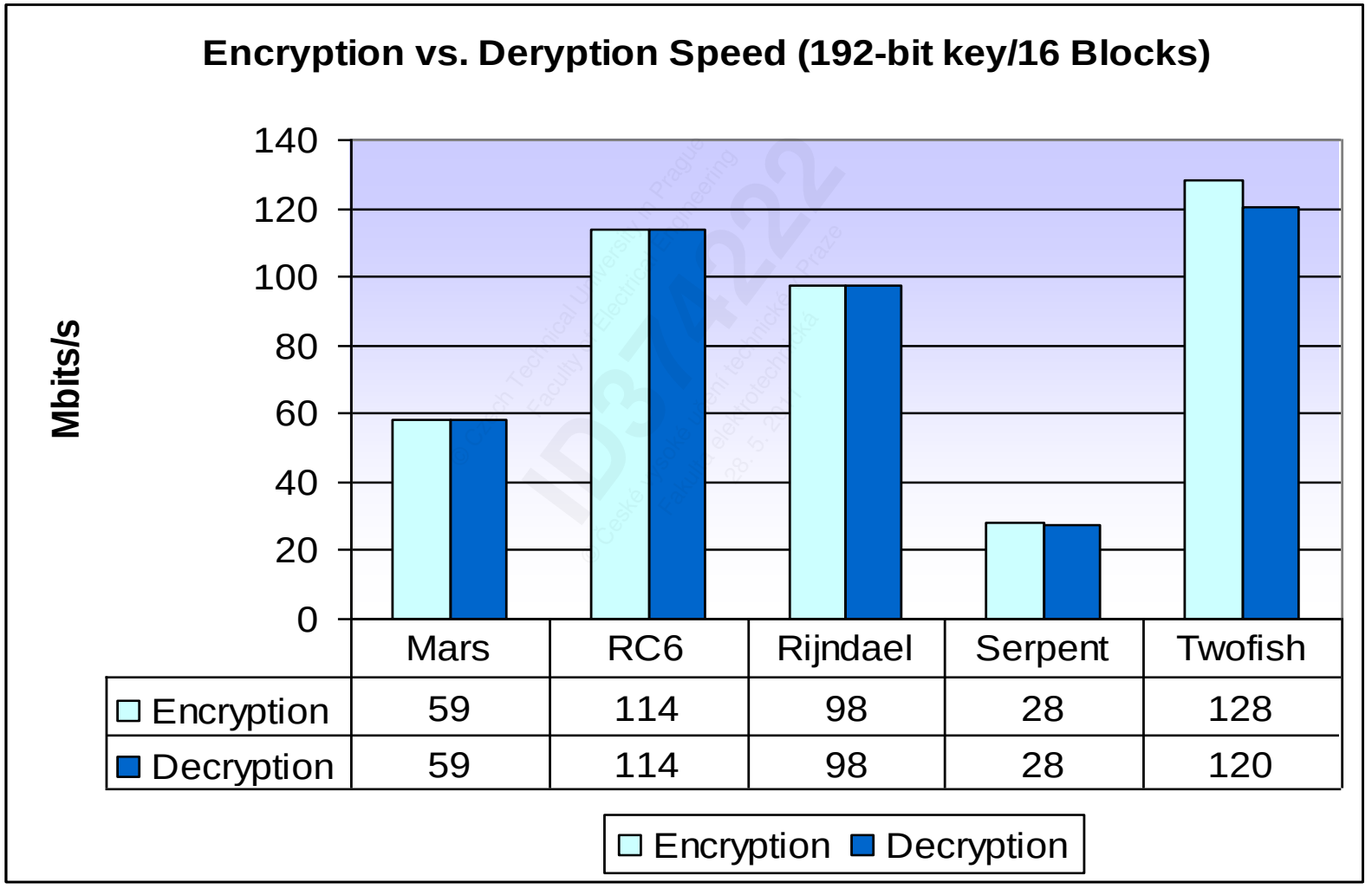
	Mars	RC6	Rijndael	Serpent	Twofish
128b	59	114	114	28	128
192b	59	114	98	28	128
256b	59	114	85	28	128

Srovnání AES a ostatních finalistů (DP)

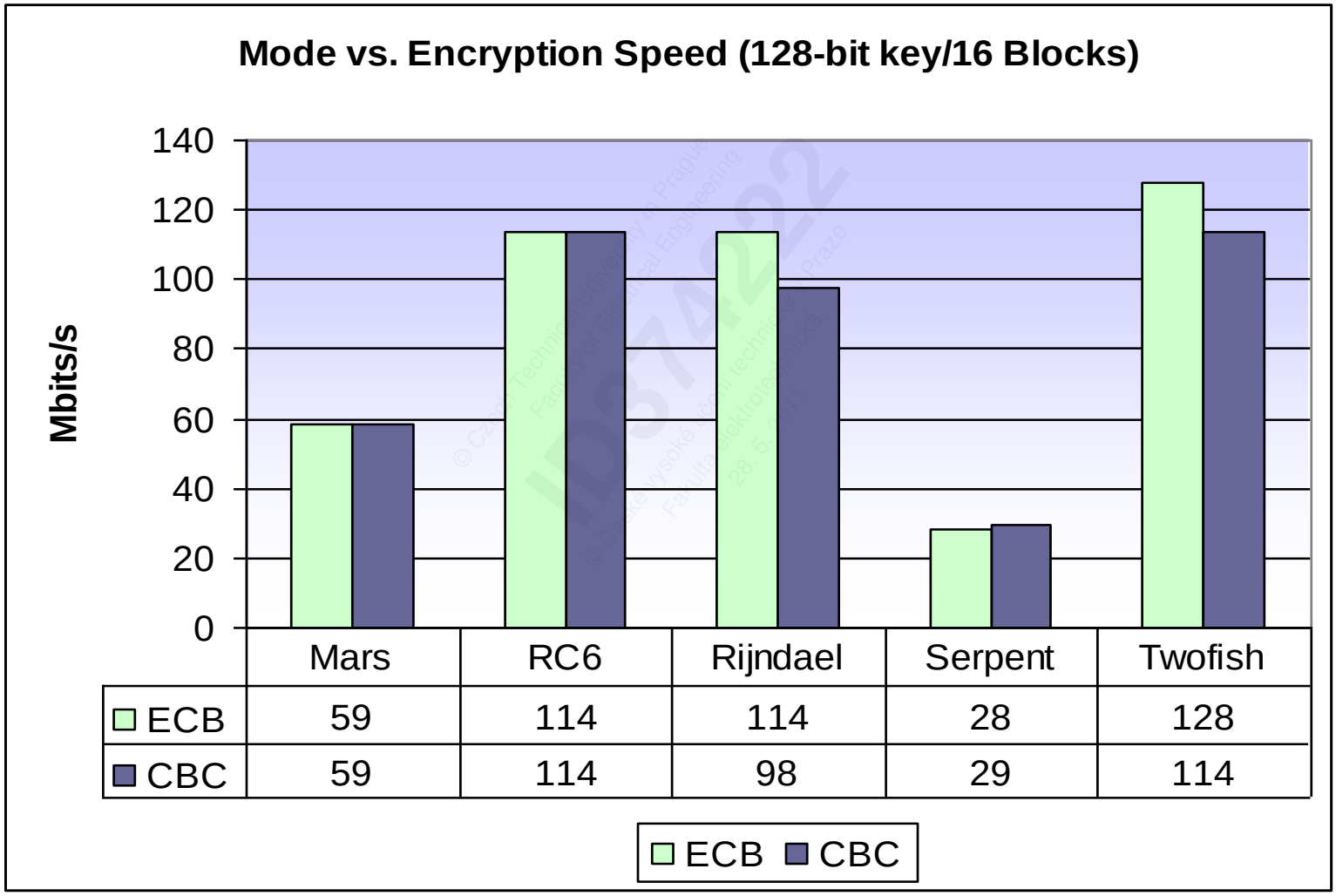


	Mars	RC6	Rijndael	Serpent	Twofish
1 Block	64	64	128	26	128
16 Blocks	59	114	98	28	128
8192 Blocks	54	102	88	25	123

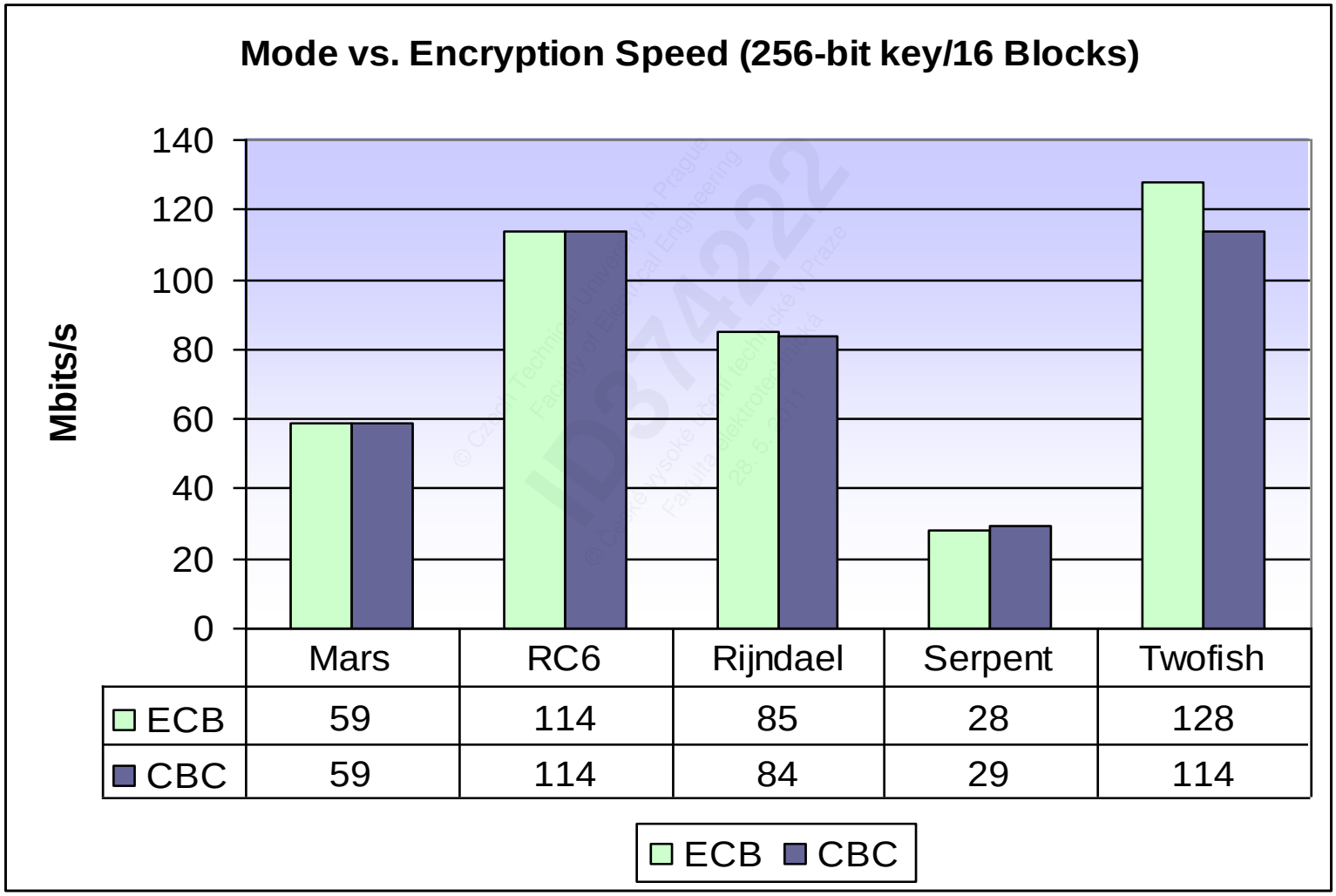
Srovnání AES a ostatních finalistů (DP)



Srovnání AES a ostatních finalistů (DP)

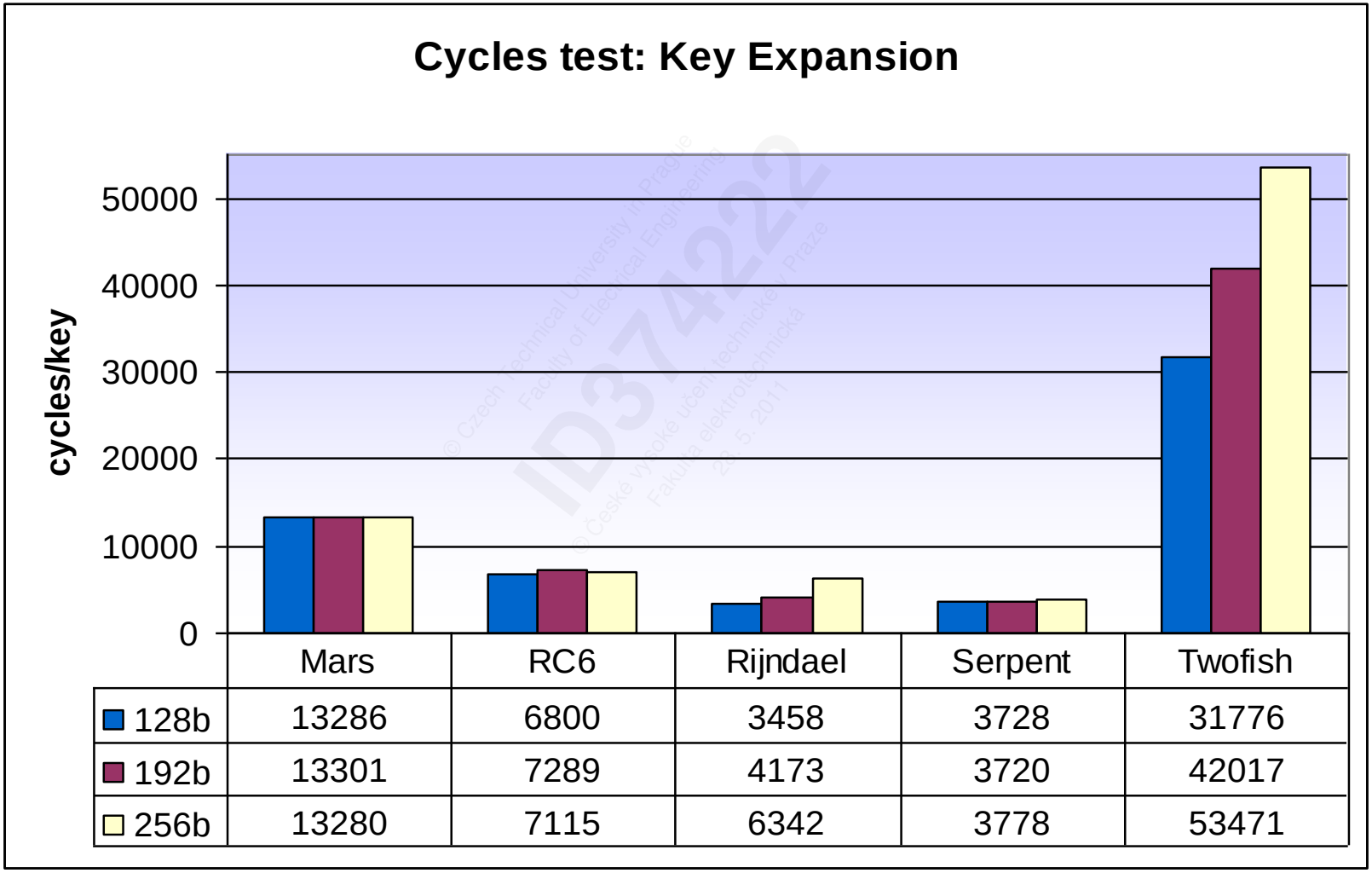


Srovnání AES a ostatních finalistů (DP)



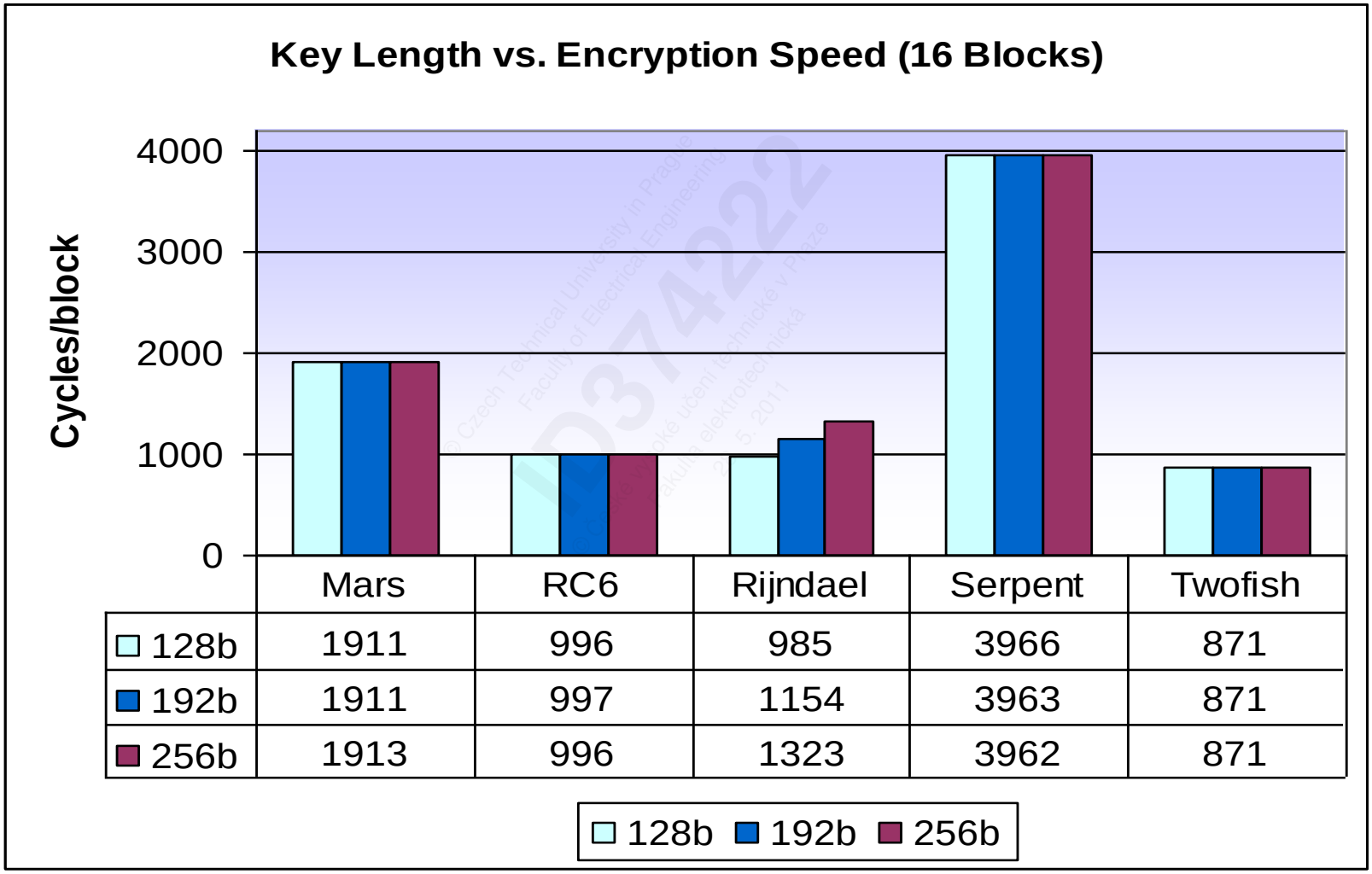
	Mars	RC6	Rijndael	Serpent	Twofish
ECB	59	114	85	28	128
CBC	59	114	84	29	114

Srovnání AES a ostatních finalistů (DP)



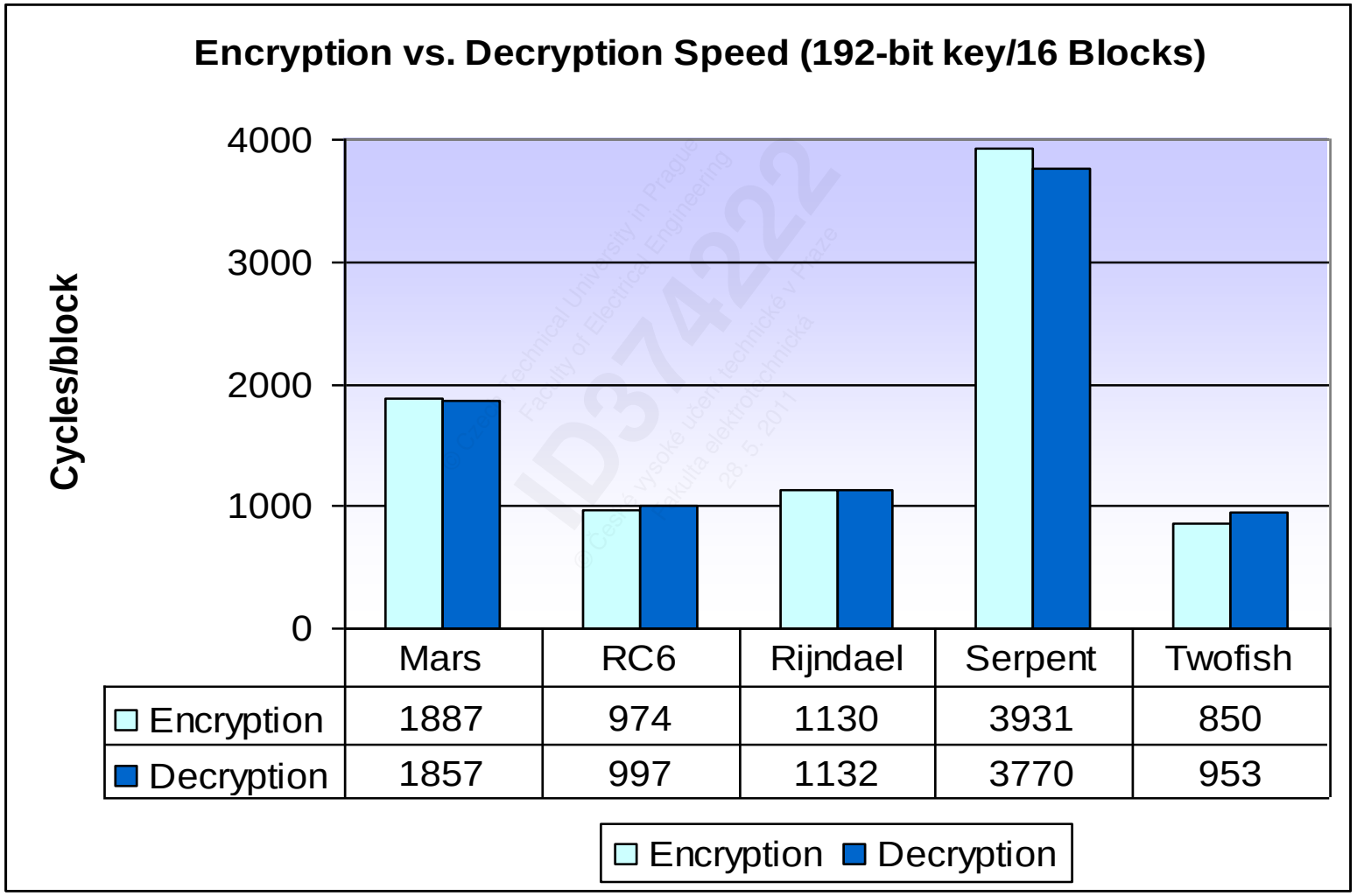
	Mars	RC6	Rijndael	Serpent	Twofish
128b	13286	6800	3458	3728	31776
192b	13301	7289	4173	3720	42017
256b	13280	7115	6342	3778	53471

Srovnání AES a ostatních finalistů (DP)

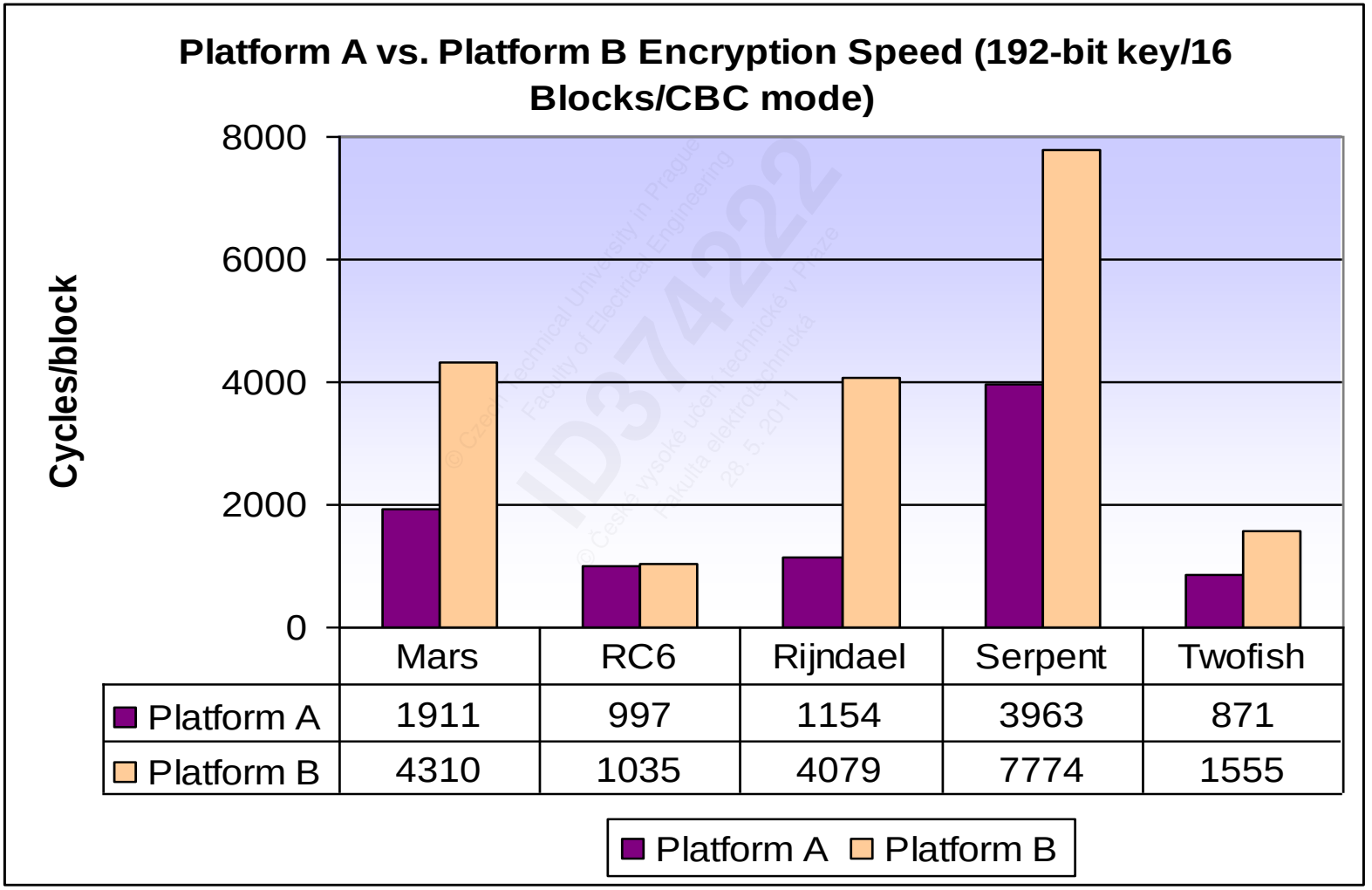


	Mars	RC6	Rijndael	Serpent	Twofish
128b	1911	996	985	3966	871
192b	1911	997	1154	3963	871
256b	1913	996	1323	3962	871

Srovnání AES a ostatních finalistů (DP)



Srovnání AES a ostatních finalistů (DP)



Zhodnocení:

Velmi malý nebo žádný vliv na výkon algoritmu má:

- režim činnosti (ECB, CBC)
- činnosti (šifrování nebo dešifrování)
- velikost vstupních dat
- velikost klíče (neplatí pro Rijndael)
- volba velikosti klíče v závislosti na rychlosti generování rundových klíčů (platí pro Mars, RC6 a Serpent)

Významný vliv na výkon byl pozorován u :

- různých velikostí klíčů pro šifrování algoritmem Rijndael
- volbu velikosti klíče v závislosti na rychlosti generování rundových klíčů (Rijndael a Twofish)
- různých CPU (Intel / AMD)

Shrnutí

MARS

+ robustní šifra
+ díky heterogenní konstrukci je i v případě zásadnějších objevů na poli krypotanalýzy nepravděpodobné, že by MARS jako celek byl postižen
+ rychlá SW implementace
- horší HW implementace
- velmi obtížná analýza kvůli složitému návrhu

RC6

+ jednoduchý návrh
+ rychlost
- SPF design
- bezpečnost (rezerva v počtu neprolomených rund)

Shrnutí

Rijndael

+ rychlý, flexibilní a „elegantní“ návrh
- možná až moc inovativní algoritmus

Serpent

+ velmi konzervativní design (počet rund)
+ bezpečnost (velký počet rund)
+ rychlost v HW
- rychlost v SW

Twofish

+ rychlost
+ flexibilita (ale na úkor „čitelnosti“ šifry)
- obtížná analýza
- složitý design (generování klíčů)

Odkazy

AES – články o vývoji algoritmu, jeho posuzování a průběhu voleb v jednotlivých kolech

http://csrc.nist.gov/archive/aes/index.html

AES Round 1 Finalists

http://csrc.nist.gov/archive/aes/round1/round1.htm#algorithms

AES Round 2 Finalists

http://csrc.nist.gov/archive/aes/round2/round2.htm#algorithms

ADVANCED ENCRYPTION STANDARD - standard

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Dotazy