# Crypto-World

**Informační sešit GCUCMP**

**ISSN 1801-2140**

**Ročník 13, číslo 6/2011** **17. červen**


# 6/2011

**Připravil: Mgr. Pavel Vondruška**

Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
**http://crypto-world.info**
(1354 registrovaných odběratelů)

Obsah :

# A. Československé šifry z obdobia 2. svetovej vojny Diel 6., Šifra „Rímska trinásť“

**Jozef Kollár,** jmkollar@math.sk
**KMaDG, SvF STU v Bratislave**

Mnohé informácie o československých šifrách z obdobia 2. svetovej vojny mi stále chýbajú. Preto ak niekto viete doplniť, prípadne opraviť mnou uvádzané popisy šifier (TTS, Rímska 2, 8, 9, 10, 13, Eva, Marta, Růžena, Utility a Palacký), alebo máte akékoľvek informácie o ďalších československých šifrách z obdobia 2. svetovej vojny, poteší ma, ak mi o tom pošlete správu.

# 6 Šifra „Rímska trinásť“

Šifra „Rímska trinásť“ bola typu TS. Jednalo sa o transpozíciu podľa obrazca a následnú polyalfabetickú substitúciu. Ako alternatívne názvy sa používali označenia ZZ5 až ZZ9, ktoré vyplývajú zo spôsobu šifrovania, čo ukážeme neskôr. Túto šifru používala napríklad radiostanica Marie v operácii Glutinium. Popis šifry je uvedený v knihe [2] (str. 123–124).

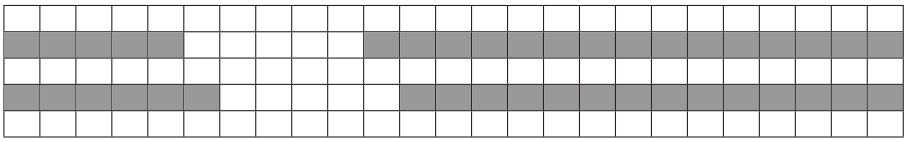
## 6.1 Všeobecný popis a príklad šifrovania depeší

Pri šifrovaní sa text naskôr transponuje podľa obrazca pripomínajúceho dve obrátené písmená „Z“ a následne sa ešte robí substitúcia. Transpozičných obrazcov je spolu päť:

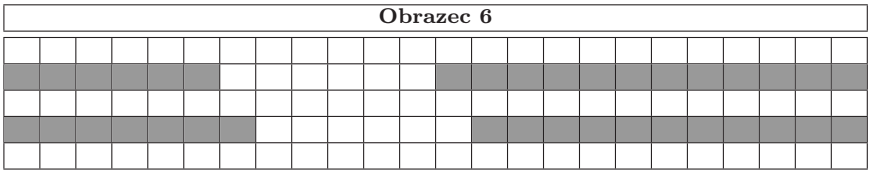
- **Obrazec 5:** 5 štvorcov – 5 štvorcov – 15 štvorcov
  Dĺžka hesla 25 znakov, počet políčok tabuľky 85.
- **Obrazec 6:** 6 štvorcov – 6 štvorcov – 12 štvorcov
  Dĺžka hesla 24 znakov, počet políčok tabuľky 84.
- **Obrazec 7:** 7 štvorcov – 7 štvorcov – 14 štvorcov
  Dĺžka hesla 28 znakov, počet políčok tabuľky 98.
- **Obrazec 8:** 8 štvorcov – 8 štvorcov – 8 štvorcov
  Dĺžka hesla 24 znakov, počet políčok tabuľky 88.
- **Obrazec 9:** 9 štvorcov – 9 štvorcov – 9 štvorcov
  Dĺžka hesla 27 znakov, počet políčok tabuľky 99.

Minimálna dĺžka hesla bola 24 znakov. Šifrovacie obrazce mali v základnej podobe 5 riadkov. Prvý, tretí a piaty riadok mali plnú šírku. Druhý riadok bol rozdelený podľa čísla obrazca tak, že prvá a tretia skupina štvorcov zostali prázdne a text sa zapisoval len do druhej skupiny štvorcov. Štvrtý riadok sa vytvoril z druhého riadku cyklickým posunom štvorcov o jeden štvorec smerom vpravo. Obrazec mal približne tvar dvoch vertikálne obrátených písmen „Z“ pod sebou (1–3 a 3–5 riadok). Transpozičné obrazce vyzerali takto:

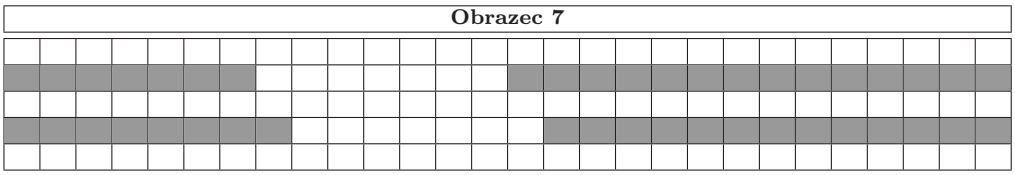
**Obrazec 5**



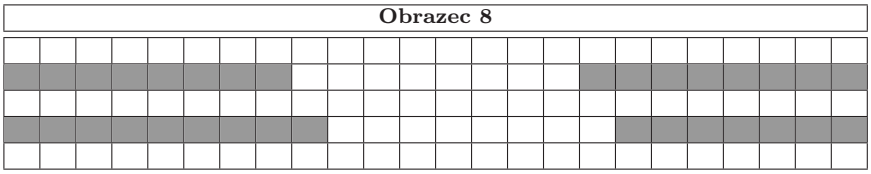
**Obrazec 6**



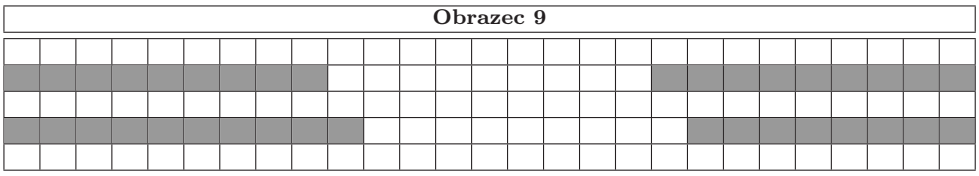
**Obrazec 7**



**Obrazec 8**



**Obrazec 9**



Text, ktorý sa išiel šifrovať, sa zapisoval pomocou 26 znakovej medzinárodnej abecedy. Dĺžne sa nepísali a spoluhlásky s mäkčeňom sa zdvojovali (napr. `Č=CC`,..,`Ř=RR`,...,`Ž=ZZ`). Ako oddeľovače sa používali dvojice znakov `QQ` alebo `XX`. Číslice a interpunkčné znamienka sa písali pomocou dvojíc znakov, podľa tzv. tabuľky `W` ([1], str. 171):

**`W` tabuľka**

| **.** | **:** | **,** | **-** | **/** | **!** | **?** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| WA | WB | WC | WD | WE | WF | WG | WH | WI | WJ | WK | WL | WM | WN | WO | WP | WQ |

Po vykonaní transpozície sa ešte zostavila substitučná tabuľka a podľa nej sa vykonala subsitúcia. Táto substitučná tabuľka mala päť riadkov. Prvý riadok sa vytvoril pomocou hlavného hesla a každý ďalší riadok bol cyklickým posunom predošlého riadku o jeden znak vľavo. Substitúcia transponovaného textu sa vykonávala po 5-členných skupinách znakov. Prvý znak sa substituoval podľa prvého riadku, druhý podľa druhého, atď. Išlo teda o polyalfabetickú substitúciu gronsfeldovho typu s nezoradenou abecedou a s periodickým heslom `12345`.

Pri šifrovaní sa využívali dve heslá. Hlavné heslo bolo pevne dané a malo mať aspoň 31 znakov (viď tvorba denného hesla). Z neho sa vytvárala substitučná tabuľka a denné heslo. Denné heslo sa z hlavného hesla vytváralo tak, že sa toto cyklicky posunulo vľavo o počet znakov zodpovedajúci dátumu šifrovania. Napr. 13. dňa v mesiaci sa denné heslo začínalo 13. znakom hlavného hesla. Denné heslo sa potom vyčíslilo obvyklým spôsobom a podľa neho sa vykonala transpozícia.

Napokon sa, podľa [2] (str. 123-124), na začiatku depeše písali tri podpisové znaky oddelené od textu bodkou. Prvý znak bol tvorený prvým znakom denného hesla, druhý znak bol jedným z dvoch znakov vpravo od prvého znaku denného hesla a tretí znak bol jedným z dvoch znakov vľavo od prvého znaku denného hesla. Takto sa to uvádza v citovanom zdroji a v príklade, ale z hľadiska šifrovania a dešifrovania to nemá žiadne opodstatnenie a je to samoúčelná komplikácia. Takýmto spôsobom robené podpisové znaky by mali zmysel jedine vtedy, ak by sa pridali nezašifrované na začiatok transponovaného textu ako 5-znaková indikačná skupina v tvare `...WA`. Je vysoko pravdepodobné, že sa to tak skutočne aj robilo a že v citovanom zdroji je chybný popis šifrovania a chybne uvedený príklad. V ďalejuvedenom príklade sa text šifruje bez podpisových znakov a tieto sú následne pridané až na začiatok transponovaného textu.

Postup šifrovania teraz ilustrujeme na príklade. Zvolíme hlavné heslo, transpozičný obrazec, dátum šifrovania, text, ktorý chceme zašifrovať a určíme si denné heslo a nadpisové znaky. Ako hlavné heslo použijeme citát Umberta Eca (1932):

> *Ak je pravý nepriateľ príliš silný,*
> *je potrebné nájsť si slabšieho.*

Za text, ktorý budeme šifrovať 13. deň v mesiaci, si zvolíme vetu:

> *V chmurných dnech roku 1940 jsme stáli zády ke zdi,*
> *střežíce pobřeží.*[1]

[1] Ryan Patrick: *Jak jsem vyhrál válku*, Naše vojsko, ESO (1985), str. 55

Na transpozíciu použijeme transpozičný obrazec číslo 8. Ten má 88 políčok, čo je presne toľko koľko potrebujeme pre zapísanie nami zvoleného textu. V praxi, ak bol text na šifrovanie dlhší, tak sa zrejme rozdeľoval na kratšie časti, ktoré sa vošli do zvolenej transpozičnej tabuľky a čiastkové depeše série sa označovali znakmi určujúcimi nadväznosť častí rovnako, ako sa to robilo pri iných šifrách. Toto je ale len náš dohad, pretože v [2] sa neuvádza ako sa nakladalo s dlhším textom a iné zdroje zatiaľ nemáme k dispozícii. Pre zvolené hlavné heslo a deň šifrovania bude denné heslo (vynechávame interpunkčné znamienka a ignorujeme medzery):

*riateľ príliš silný je potrebné nájsť si slabšieho Ak je pravý nep*

Denné heslo sa začína 13. písmenom hlavného hesla. Nadpisové znaky budú potom napríklad `RIE`, ale mohli by byť aj `RIP` atď. Spolu máme štyri možnosti ich výberu.

Nakreslíme transpozičný obrazec, zapíšeme doň otvorený text v upravenom tvare a vyčíslime denné heslo, podľa ktorého vykonáme transpozíciu. Vyčísleným denným heslom očíslujeme stĺpce transpozičného obrazca (tabuľky):

**Obrazec 8**

| R | I | A | T | E | L | P | R | I | L | I | S | S | I | L | N | Y | J | E | P | O | T | R | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 17 | 5 | 1 | 22 | 2 | 10 | 15 | 18 | 6 | 11 | 7 | 20 | 21 | 8 | 12 | 13 | 24 | 9 | 3 | 16 | 14 | 23 | 19 | 4 |
| V | X | X | C | H | M | U | R | N | Y | C | H | Q | Q | D | N | E | C | H | X | X | R | O | K |
| | | | | | | | | U | Q | Q | W | I | W | Q | W | | | | | | | | |
| L | W | H | X | X | J | S | M | E | Q | Q | S | T | A | L | I | X | X | Z | A | D | Y | Q | Q |
| | | | | | | | | | K | E | X | X | Z | D | I | W | | | | | | | |
| C | S | T | R | R | E | Z | Z | I | C | E | Q | Q | P | O | B | R | R | E | Z | Z | I | W | A |

Z transpozičnej tabuľky budeme vypisovať text po stĺpcoch zhora nadol, v poradí určenom vyčísleným denným heslom. Tento text zároveň rozdelíme do 5-znakových skupín:

```
XHTHX RHZEK QAXWS NUEIC QQEEQ WAZPC XRMJE YQQKC DQLDO NWIIB
XDZUS ZXAZV LCRMZ OQWHW SXQQI TXQCX RRYIE XWR
```

Ďalej na začiatok transponovaného textu pridáme nadpisové znaky ako indikačnú skupinu:

```
RIEWA XHTHX RHZEK QAXWS NUEIC QQEEQ WAZPC XRMJE YQQKC DQLDO
NWIIB XDZUS ZXAZV LCRMZ OQWHW SXQQI TXQCX RRYIE XWR
```

O služobných údajoch a ich umiestňovaní v depeši sa v [2] nepíše. Každá depeša ale určite musela obsahovať záhlavie aj služobné údaje, aby bola dešifrovateľná. Záhlavie muselo obsahovať dátum šifrovania, pretože podľa neho sa

vykonával posun denného hesla a služobné údaje museli obsahovať informáciu o použitej transpozičnej tabuľke. Okrem toho zrejme záhlavie obsahovalo aj poradové číslo a počet znakov depeše a služobné údaje obsahovali zrejme aj informáciu o nasledujúcej relácii. Takže pre náš príklad doplníme záhlavie depeše rovnakým spôsobom, ako sa to robilo aj pri väčšine ďalších šifier. Záhlavie bude v tvare `xxx-yyy-zz`, kde `xxx` je poradové číslo depeše, `yyy` je počet cifier depeše a `zz` je deň šifrovania depeše. O formáte služobných údajov nebudeme špekulovať, pretože o nich absolútne nič nevieme. Jednoducho teda na koniec zašifrovanej depeše zapíšeme číslo použitej transpozičnej tabuľky (5 až 9) vo formáte podľa tabuľky `W` uvedenej na strane 3. Pokiaľ počet znakov depeše nebude násobok 5, tak ešte na záver doplníme potrebný počet znakov `Q` a `X` náhodným spôsobom. Je jasné, že takto reálne služobné údaje určite neboli kódované, pretože by boli príliš ľahko identifikovateľné. Vzhľadom na chýbajúce informácie to ale v našom príklade budeme robiť týmto spôsobom. Ak teda transponovanú depešu z nášho príkladu doplníme o služobné údaje, tak dostávame:

```
RIEWA XHTHX RHZEK QAXWS NUEIC QQEEQ WAZPC XRMJE YQQKC DQLDO
NWIIB XDZUS ZXAZV LCRMZ OQWHW SXQQI TXQCX RRYIE XWRWP
```

Na základe hlavného hesla vytvoríme substitučnú tabuľku a podľa nej substituujeme transponovaný text aj s indikačnou skupinou na začiatku. Táto substitučná tabuľka bude mať päť riadkov a obsahuje len 26 znakov medzinárodnej abecedy. Prvý riadok dostaneme zapísaním abecedy v poradí určenom hlavným heslom a každý ďalší riadok bude cyklickým posunom predošlého riadku o jeden znak doľava. V našom príklade dostávame tabuľku 1 uvedenú na strane 7.

Následne vykonáme substitúciu transponovaného textu podľa takto skonštruovanej substitučnej tabuľky. Je to polyalfabetická substitúcia gronsfeldovho typu s heslom `12345`, t.j. v každej pätici znakov sa prvý znak nahrádza podľa prvého riadku, druhý znak podľa druhého riadku atď.

V našom príklade zvolíme číslo depeše 45 a po vykonaní substitúcie dostávame definitívne zašifrovaný text:

```
045-095-13
QIVGP DNZTK QNKYB MKGGC OZVLV MQVYX CKKUV DUBSN FQUOV EQOVU
ODTLR DPKDC GFJJG LPWHE BQFTA UFUWS WFURK QUALN DDWGW
```

ktorý je týmto pripravený na odoslanie.

```
   A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
*****************************************************
1 | A K J E P R V Y N I T L S O B H M Q U W X Z C D F G
--|--------------------------------------------------
2 | K J E P R V Y N I T L S O B H M Q U W X Z C D F G A
--|--------------------------------------------------
3 | J E P R V Y N I T L S O B H M Q U W X Z C D F G A K
--|--------------------------------------------------
4 | E P R V Y N I T L S O B H M Q U W X Z C D F G A K J
--|--------------------------------------------------
5 | P R V Y N I T L S O B H M Q U W X Z C D F G A K J E
---------------------------------------------------
```

Tabuľka 1: Substitučná tabuľka pre „rímsku trinásť“

## 6.2 Postup pri šifrovaní

V tejto časti budeme vychádzať z nasledovných predpokladov:

a. Máme k dispozícii text na šifrovanie.

b. Máme dané aspoň 31 znakov dlhé hlavné heslo.

c. Je daný deň šifrovania.

d. Zvolíme obrazec použitý na transpozíciu (5 až 9).

Potom šifrovanie depeše bude prebiehať v nasledovných krokoch:

1. Text, ktorý ideme šifrovať prepíšeme len pomocou 26 znakov medzinárodnej abecedy. Dĺžne sa nepíšu a spoluhlásky s mäkčeňom sa zapisujú zdvojene a bez mäkčeňa (napr. `Š=SS, Ť=TT,...`).

2. Interpunkčné znamienka `. : , - / ! ?` a cifry zapisujeme pomocou dvojíc znakov, podľa tzv. `W` tabuľky, uvedenej na strane 3.

3. Medzery sa nahrádzajú dvojicou znakov `QQ` alebo `XX` a za špeciálnymi znakmi sa medzery nepíšu, podobne ako tomu bolo aj pri iných šifrách.

4. Podľa toho, ktorý transpozičný obrazec (5 až 9) sme zvolili, rozdelíme text na kratšie časti. Počet políčok jednotlivých transpozičných tabuliek sa pohybuje od 84 do 99 (viď strana 2). Do tohto počtu treba zarátať aj znaky zabezpečujúce nadväznosť častí a treba dbať na to, aby žiadne dve časti nemali po rozdelení a zakódovaní rovnakú dĺžku.

5. Na koniec prvej časti pridáme, kvôli nadväznosti dielov `/A`. Na začiatok druhej časti pridáme `A/`, na koniec druhej časti pridáme `/B` atď. Každá časť textu (okrem prvej a poslednej) bude mať na začiatku písmeno identické s koncovým písmenom predošlej časti a znak `/` a na konci textu znak `/` a písmeno identické s písmenom označujúcim nasledovnú časť textu. Písmena na označovanie častí berieme podľa abecedy. Prvá časť má označenie len na konci a posledná časť len na začiatku.

6. Každú časť textu šifrujeme zvlášť a každá časť textu tvorí samostatnú depešu. Ďalší popis sa bude týkať šifrovania jednotlivých častí.

7. Rozdelenú a zakódovanú depešu doplnenú o znaky zabezpečujúce nadväznosť zapíšeme po riadkoch do zvolenej transpozičnej tabuľky. Znaky zapisujeme zľava doprava a zhora nadol.

8. Nad stĺpce transpozičnej tabuľky napíšeme denné heslo, ktoré je cyklickým posunom hlavného hesla. Začína sa tým písmenom hlavného hesla, ktorého poradie zodpovedá dňu šifrovania.

9. Obvyklým spôsobom vyčíslime denné heslo. Medzery a interpunkčné znamienka sa nevyčíslujú a znaky vyčíslujeme podľa 26 znakovej medzinárodnej abecedy.

10. Znaky z transpozičnej tabuľky vypisujeme po stĺpcoch zhora nadol a stĺpce berieme v poradí určenom vyčíslením denného hesla. Znaky zapisujeme v päťmiestnych skupinách.

11. Pokiaľ robíme transpozíciu prvej časti série, tak na začiatok už transponovaného textu následne pridáme tri podpisové znaky a bodku, t.j. päticu znakov v tvare `...WA`, kde `...` sú tri podpisové znaky. Podpisové znaky sa tvorili na základe denného hesla. Prvý znak podpisu bol prvým znakom denného hesla, druhý znak podpisu bol jedným z dvoch znakov vpravo od prvého znaku denného hesla a tretí znak podpisu bol jedným z dvoch znakov vľavo od prvého znaku denného hesla.

12. Na koniec transponovanej depeše doplníme služobné údaje. **Nevieme ako sa v skutočnosti tvorili skupiny služobných údajov. Tu popisovaný postup je vymyslený! Služobné údaje museli obsahovať informáciu o použitom transpozičnom obrazci (tabuľke) a pravdepodobne obsahovali deň mesiaca, hodinu a minútu nasledujúcej relácie.** V našom prípade budú služobné údaje pozostávať len z čísla použitej transpozičnej tabuľky (5 až 9) vo formáte podľa `W` tabuľky uvedenej na strane 3. Ak počet znakov depeše nie je násobok

5, tak na jej koniec ešte náhodným spôsobom doplníme potrebný počet znakov `Q` a `X`.

13. Podľa hlavného hesla vytvoríme päťriadkovú substitučnú tabuľku. Táto bude obsahovať len 26 znakov medzinárodnej abecedy. V prvom riadku tabuľky bude medzinárodná abeceda zapísaná v poradí určenom hlavným heslom obvyklým spôsobom. V každom ďalšom riadku tabuľky bude abeceda z predošlého riadku cyklicky posunutá o jeden znak vľavo.

14. Spravíme polyalfabetickú substitúciu už transponovaného textu, v prípade prvej časti série doplnenej aj o podpisové znaky. Túto substitúciu robíme podľa tabuľky skonštruovanej v bode 13 tak, že v každej pätici znakov substituujeme prvý znak podľa prvého riadku tabuľky, druhý znak podľa druhého riadku tabuľky atď.

15. Na začiatok depeše pridáme ešte návestie v tvare `xxx-yyy-zz`, kde `xxx` je poradové číslo depeše, `yyy` je počet cifier depeše a `zz` je deň šifrovania depeše. Týmto je šifrovanie depeše ukončené a depeša je pripravená na odoslanie.

## 6.3 Postup pri dešifrovaní

V tejto časti budeme vychádzať z nasledovných predpokladov:

a. Máme k dispozícii kompletný text zašifrovanej depeše.

b. Máme dané hlavné heslo.

Potom dešifrovanie depeše bude prebiehať v nasledovných krokoch:

1. Na základe návestia overíme kompletnosť depeše (počet cifier).

2. Z návestia zistíme deň šifrovania a podľa neho potom z hlavného hesla dostaneme denné heslo. Denné heslo bude cyklickým posunom hlavného hesla a začína sa tým jeho znakom, ktorý zodpovedá dňu šifrovania.

3. Vynecháme návestie depeše, ktoré už nebudeme potrebovať.

4. Podľa hlavného hesla vytvoríme päťriadkovú substitučnú tabuľku. Táto bude obsahovať len 26 znakov medzinárodnej abecedy. V prvom riadku tabuľky bude medzinárodná abeceda zapísaná v poradí určenom hlavným heslom obvyklým spôsobom. V každom ďalšom riadku tabuľky bude abeceda z predošlého riadku cyklicky posunutá o jeden znak vľavo.

5. Podľa substitučnej tabuľky z predošlého bodu dešifrujeme depešu. Pri šifrovaní sme použili polyalfabetickú substitúciu gronsfeldovho typu s heslom `12345`. Takže pri dešifrovaní v každej pätici znakov prvý znak dešifrujeme podľa prvého riadku tabuľky, druhý podľa druhého atď.

6. Ak sú na konci dešifrovanej depeše znaky `Q` alebo `X` (môžu tam byť 1 až 4 takéto znaky), vynecháme ich. Sú to len nuly doplňujúce počet znakov depeše na násobok 5.

7. Posledné dva znaky depeše sú číslo zakódované pomocou `W` tabuľky zo strany 3. Toto číslo je v rozsahu 5 až 9 a určuje nám, ktorý obrazec bol použitý pri transpozícii. Po získaní tejto informácie posledné dva znaky opäť vynecháme.

8. Ak sa jedná o prvú časť série, na začiatku by mali byť podpisové znaky. Je to pätica znakov v tvare `...WA`, kde `...` sú tri podpisové znaky. Podpisové znaky sa tvorili na základe denného hesla. Prvý znak podpisu bol prvým znakom denného hesla, druhý znak podpisu bol jedným z dvoch znakov vpravo od prvého znaku denného hesla a tretí znak podpisu bol jedným z dvoch znakov vľavo od prvého znaku denného hesla. Podľa tohto spoznáme prvú časť série a túto prvú päticu znakov z prvej časti série vynecháme[2].

9. Nad stĺpce transpozičnej tabuľky napíšeme denné heslo, ktoré je cyklickým posunom hlavného hesla. Začína sa tým písmenom hlavného hesla, ktorého poradie zodpovedá dňu šifrovania.

10. Obvyklým spôsobom vyčíslime denné heslo. Medzery a interpunkčné znamienka sa nevyčíslujú, znaky vyčíslujeme podľa 26 znakovej medzinárodnej abecedy.

11. Dĺžku depeše (už bez prípadných podpisových znakov na začiatku a služobných údajov na konci) poznáme. Takisto vieme, ktorý transpozičný obrazec sa použil a koľko má políčok. Určíme teda, ktoré riadky transpozičnej tabuľky budú obsadené a ako dlhé budú jej jednotlivé stĺpce.

12. Znaky depeše zapisujeme po stĺpcoch zhora nadol do transpozičnej tabuľky. Stĺpce zapisujeme v poradí určenom vyčísleným denným heslom a dĺžky jednotlivých stĺpcov sme určili v predchádzajúcom bode.

[2]Popis v [2] nie je správny a aj táto nami upravená verzia tvorby podpisových znakov je zbytočná a samoúčelná.

13. Text vypisujeme z transpozičnej tabuľky po riadkoch zľava doprava a zhora nadol. Text je písaný len pomocou 26 znakov medzinárodnej abecedy.

14. Niektoré interpunkčné znamienka a cifry sú kódované podľa `W` tabuľky uvedenej na strane 3. Zapíšeme ich teda v pôvodnej podobe.

15. Dvojice `QQ` a `XX` predstavujú medzery a iné zdvojené spoluhlásky predstavujú príslušnú spoluhlásku s mäkčeňom. Zapíšeme teda tieto znaky v ich pôvodnej podobe.

16. Doplníme medzery za špeciálne znaky v texte. Týmto sme dostali pôvodný text depeše.

17. Pokiaľ sa jedná o sériu, tak text zostavíme v správnom poradí podľa označenia na začiatku a konci jednotlivých častí série.

### 6.4 Lúštenie

Táto šifra je typu TS, pričom substitúcia je gronsfeldovho typu s veľmi krátkym heslom `12345`. Na veci nič nemení ani fakt, že substitučná abeceda je neusporiadaná. Navyše na tvorbu substitučnej abecedy a na transpozíciu sa použilo rovnaké heslo. Pokiaľ ide o transpozíciu, tak ak máme viacero rovnako dlhých textov šifrovaných tým istým heslom a pomocou tej istej transpozičnej tabuľky, tak môžme použiť anagramovú metódu.

## Literatúra

[1] Grošek Otokar, Vojvoda Milan, Zajac Pavol: Klasické šifry
*STU v Bratislave, 2007*

[2] Hanák Vítězslav: Muži a radiostanice tajné války
*Elli Print, 2002*

[3] Janeček Jiří: Gentlemani (ne)čtou cizí dopisy
*Books Bonus A, 1998*

[4] Janeček Jiří: Odhalená tajemství šifrovacích klíčů minulosti
*Naše vojsko, 1994*

[5] Janeček Jiří: Válka šifer – výhry a prohry československé vojenské rozvědky (1939–1945)
*Votobia, 2001*

# B. Kryptografický softwarový nástroj CipherCAD a kryptoanalýza

**Vlastimil Klíma (vlastimil.klima@knzsro.cz, KNZ, s.r.o. Praha),**
**Václav Plátěnka ( vaclav.platenka@unob.cz , Univerzita obrany Brno)**

Tento článek je z velké části tvořen překladem společného příspěvku Klima-Platenka na SPI 2011[5], navíc je doplněný o videa z prezentace k tomuto příspěvku.

CipherCAD je grafický softwarový programovací nástroj, který může být využit k modelování a zkoumání kryptografických funkcí, protokolů apod. V tomto příspěvku prezentujeme použití CipherCADu k definici a zkoumání pěti finalistů soutěže NIST SHA-3. Na omezeném prostoru ukazujeme několik z nepřeberného množství možností využití CipherCADu ke kryptoanalýze a zkoumání kryptografických funkcí. Možnosti tvorby kryptoschémat ukazujeme na příkladu hašovací funkce Skein-512. Poté je představena srovnávací analýza na testu lavinovitosti pro všech pět finalistů SHA-3: BLAKE, Grøstl, JH, KECCAK a Skein.

## 1 Úvod

CipherCAD byl vytvořen v rámci výzkumu a vývoje pro Národní bezpečnostní úřad [1]. Všechny studované kryptografické funkce nebo protokoly jsou v CipherCADu sestaveny ze stavebních bloků, které mohou být libovolně vnořovány, řetězeny a spojovány. CipherCAD umožňuje dívat se na tok dat, data ukládat nebo s nimi provádět další testy.

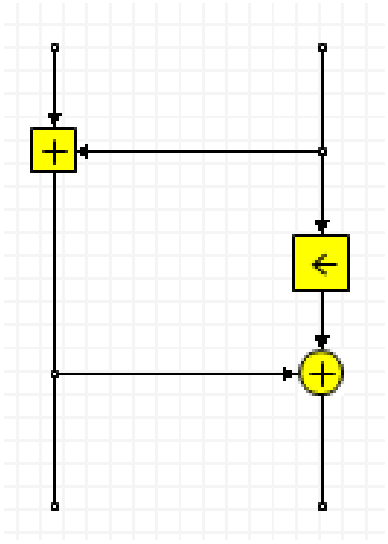
CipherCAD je softwarová aplikace, která je určena pro analýzu a syntézu kryptografických algoritmů a protokolů. Lze v ní realizovat funkční modely všech současných symetrických a asymetrických kryptografických algoritmů. CipherCAD také obsahuje analytické nástroje umožňující provádět statistické a algebraické testy kryptografických algoritmů a jejich stavebních bloků. V rámci aplikace byla zpracována více než stovka kryptografických algoritmů, analytických postupů a standardizovaných protokolů. Tyto jsou přístupny v „knize schémat“ nebo jako samostatné „Sešity“ a jsou dostupné z titulní stránky aplikace. Aplikace také podporuje tvorbu dokumentace kryptografických algoritmů a protokolů.

CipherCAD poskytuje pro tvorbu matematických modelů grafické interaktivní rozhraní, které odpovídá současné úrovni aplikací typu CAD (Computer Aided Design). Uživatel vytváří funkční modely kryptografických prvků téměř výhradně grafickými prostředky a použitím vybraných komponent. Aplikace umožňuje generovat potřebná data k algebraické i statistické analýze, a to z libovolné části schématu nebo z jeho libovolné komponenty.

## 2 Model algoritmu Skein v CipherCADu

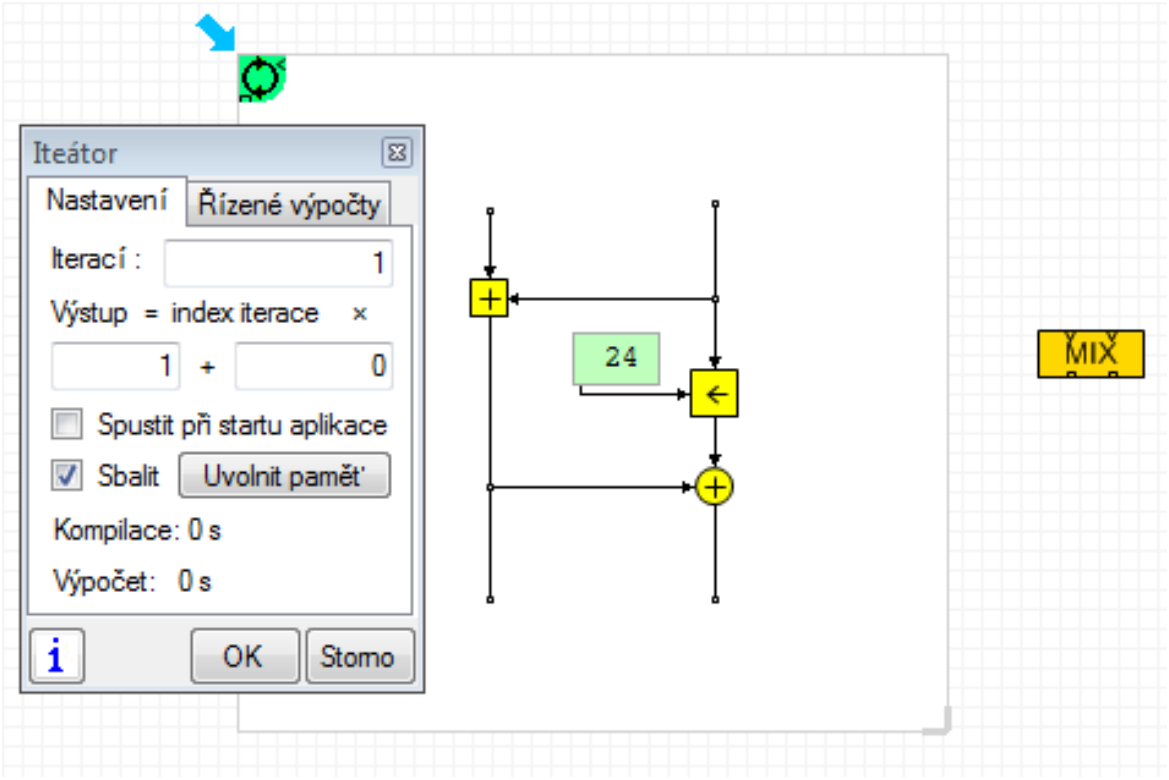
Pro demonstraci tvorby modelu kryptografické funkce jsme zvolili hašovací algoritmus Skein-512-512. Jeho jádrem je tweakovatelná bloková šifra Threefish-512 s klíčem o 512 bitech a tweakem o 128 bitech. Threefish využívá operace XOR, ADD a ROT (o konstantní počet bitů) se 64-bitovými slovy. Z těchto operací je sestavena základní nelineární část šifry Threefish nazvaná MIX, viz obr. 1.

Sestavování provádíme pomocí kontextového menu a spojovacích čar.



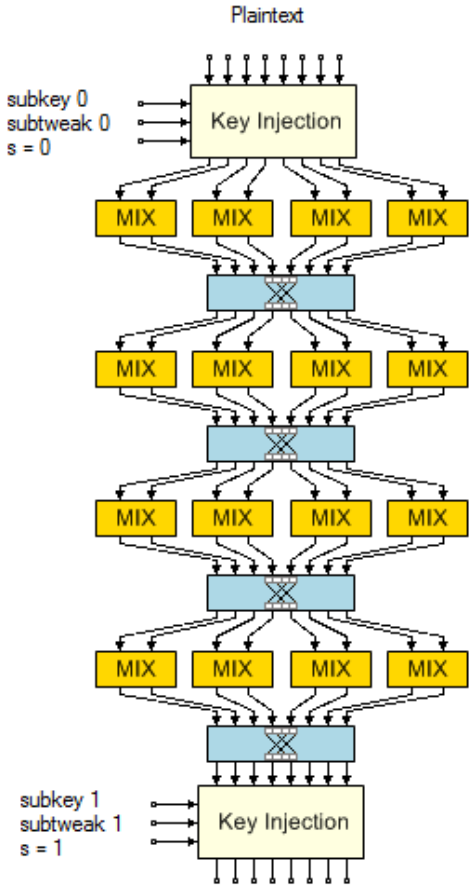
Obr. 1: Pomocí spojů a komponent vzniká logická struktura MIX

Na následujícím obrázku vidíme „zapouzdření“ této struktury do tzv. iterátoru. Tím vytvoříme samostatný funkční blok. Iterátor můžeme „sbalit“, čímž vznikne nový malý (grafický) blok, ktrerý označíme „MIX“. MIX je tedy nová funkce, která má dva 64-bitové vstupy a dva 64-bitové výstupy, viz obr. 2 vpravo. Postup tvorby komponenty MIX můžete sledovat na videu schema.mpg (55,3 MB, http://crypto-world.info/cw6/schema.mpg [6] ).



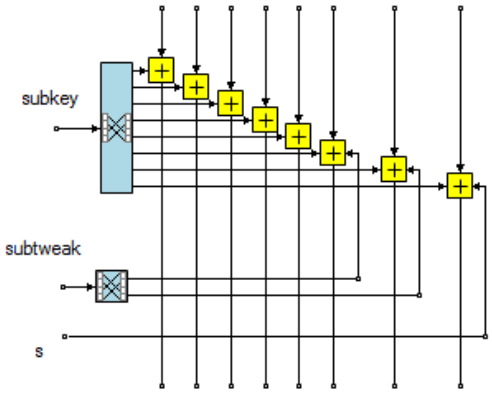

Obr. 2: Iterátor, zapouzdření a vznik nové funkce MIX

Bloková šifra Threefish-512 pracuje v 72 rundách, každá z těchto rund se skládá ze čtyř fukcí MIX a jedné permutace osmi 64-bitových slov, viz obr. 3.




Obr. 3: Čtyři ze 72 rund blokové šifry Threefish-512

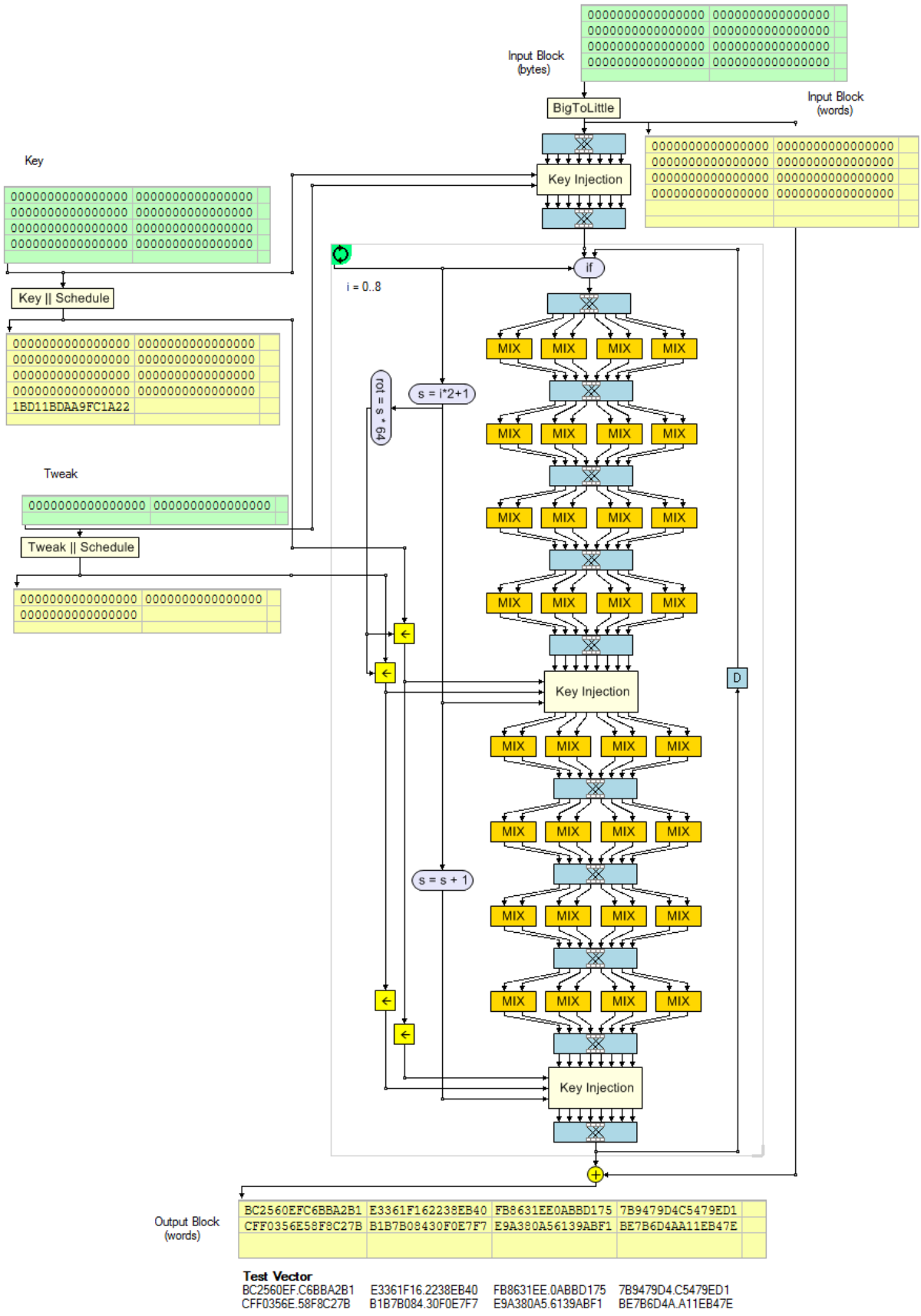
Vždy po čtyřech rundách je na data přičten příslušný podklíč, jak ukazují obrázky 3 a 4.




Obr.4: Vkládání klíče v Threefish-512

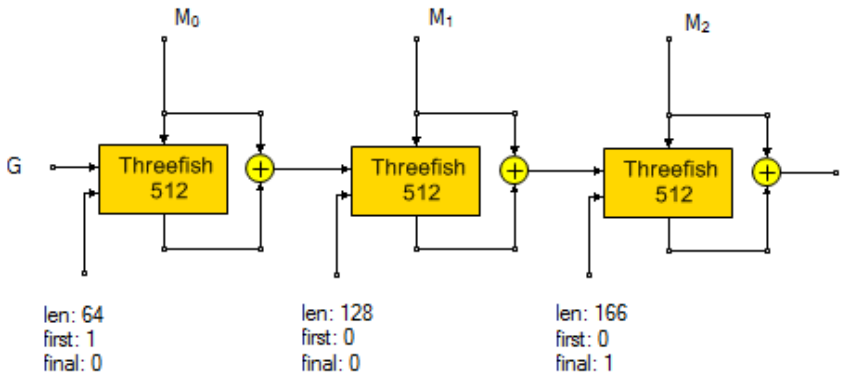
Na obr. 5 je znázorněn celý algoritmus Threefish s přídavným závěrečným přixorováním otevřeného textu. Hlavní částí Threefish je iterátor, který obsahuje osm rund a je volán devětkrát. To dává 72 rund Threefish.

Výstup z jedné iterace je veden na vstup další iterace pomocí paměťové komponenty „zpětná vazba D“.



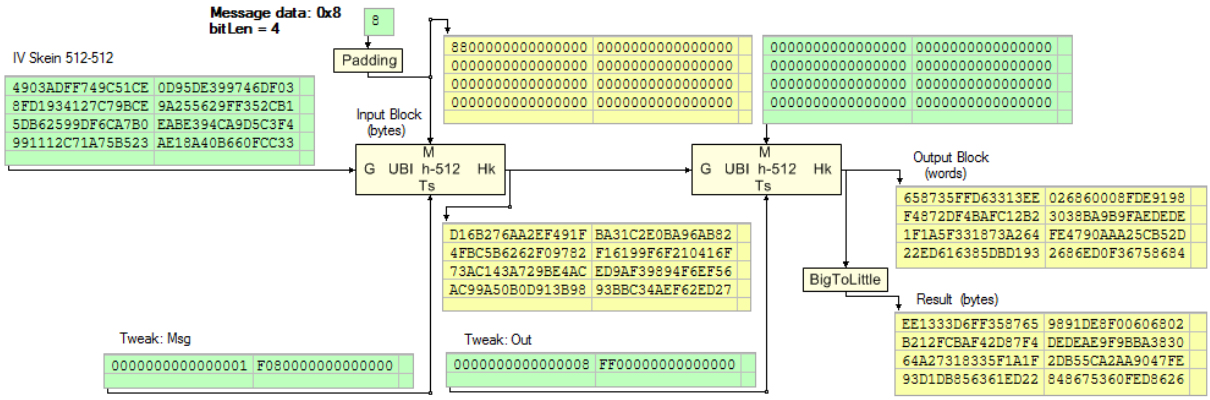

Obr.5: Jeden blok UBI (Threefish se závěrečným přixorováním vstupních dat)

Threefish se závěrečným přixorováním otevřeného textu je základní jádro bloku UBI, jak je znázorněno na obr. 6. UBI zpracovává mnoho bloků zprávy jednoduše tak, že volá jádro UBI mnohokrát za sebou, jak ukazuje obr. 6. Výsledek jádra UBI plní vždy do proměnné klíče v následujícím jádru UBI a další blok zprávy plní do vstupního bloku UBI. Tímto řetězením je pomocí UBI zpracována zpráva libovolné délky. Na obrázku 6 je příklad výpočtu Skein-512 pro 166 bajtový vstup.




Obr. 6: Při hashování 166 bajtů volá Skein-512 třikrát UBI

Blok UBI opět můžeme sbalit do jednoho stavebního prvku. Můžeme to vidět na obrázku 7, společně s hodnotami vstupu a výstupu bloku UBI.

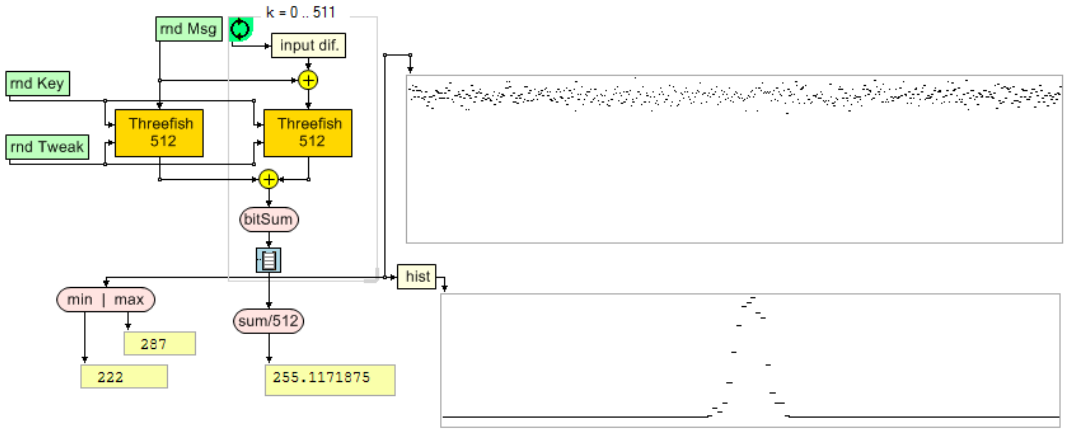



Obr. 7: Zobrazení průběžných hodnot UBI při hašování zprávy „1000“ o délce 4 bity

## 3 Kryptoanalýza v CipherCADu

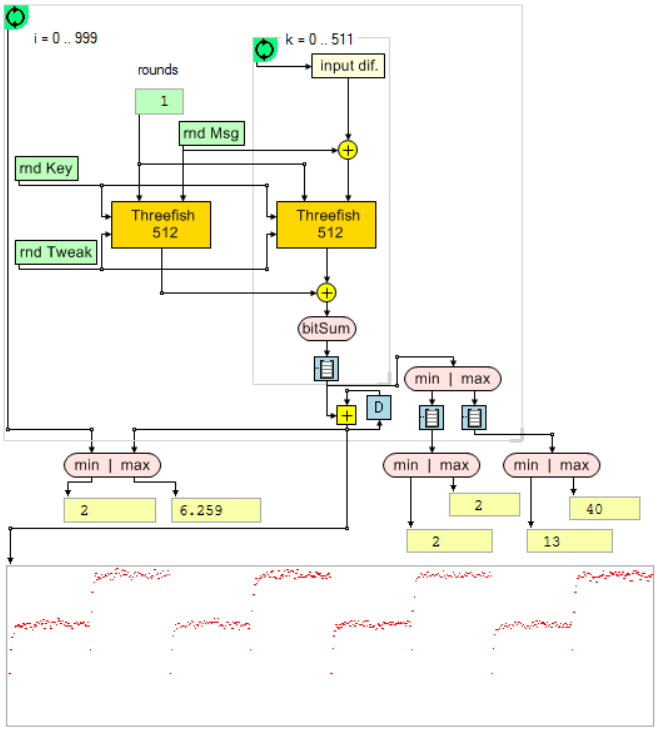
Po vytvoření modelu algoritmu můžeme za pomoci CipherCADu provádět různá zkoumání algoritmu. Jeden ze základních testů u blokových šifer je test lavinovitosti, který testuje, zda změna jednoho bitu na vstupu vede ke změně každého bitu na výstupu s pravděpodobností 0.5. Aby bylo možno otestovat vliv každého jednotlivého bitu na vstupu blokové šifry, postupujeme s měnícím se bitem od nejméně významného až po nejvíce významný bit vstupního bloku (metoda kráčejícího bitu). Nyní pro každou vstupní pozici vypočítáme, kolik se změnilo bitů výstupního bloku. Výsledek zobrazíme v grafu, kde na ose x je pozice vstupní jednobitové diference a na ose y je počet bitových změn na výstupu. Na obr. 8 je test

lavinovitosti pro celou blokovou šifru Threefish-512 se 72 rundami. Z výsledků je vidět, že nedochází k významným odchylkám od předpokládaných parametrů a z histogramu vidíme, že se změní vždy kolem poloviny výstupních bitů.




Obr. 8: Test lavinovitosti metodou kráčejícího bitu pro blokovou šifru Threefish-512

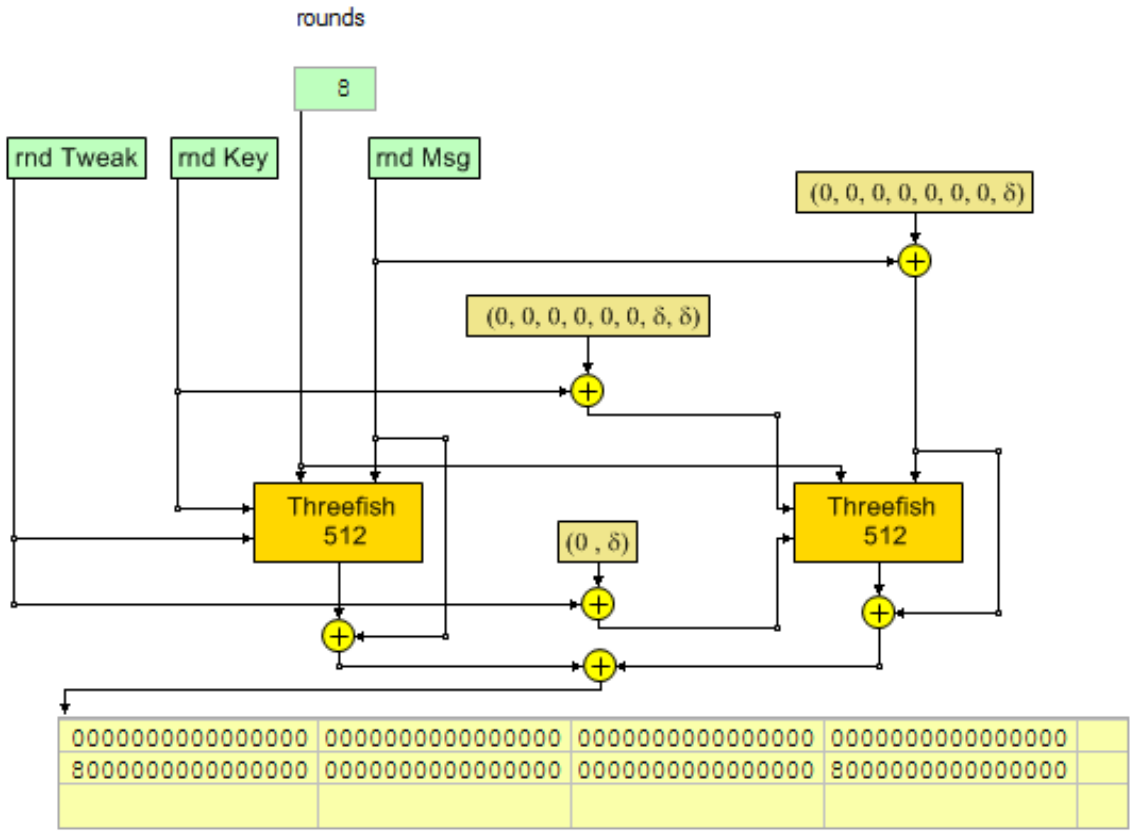
Na obr. 9 je naproti tomu zobrazen test kráčejícího bitu pro jednu rundu algoritmu Threefish-512, čili byl proveden první key injection, čtyři funkce MIX a jedna permutace osmi 64-bitových slov. Pro lepší pochopení vlivu konkrétního bitu, je celý test opakován 1000 krát s různou hodnotou náhodného vstupního bloku a výsledky jsou zprůměrovány. Z grafu na obr. 9 je vidět vliv struktury nelineární funkce MIX, na difuzi vstupní diference. Je vidět závislost na tom, zda vstupní diference je v levém nebo pravém vstupním slově funkce MIX. Z levého obvodu hledajícího minimum a maximum vyplývá, že minimální průměrná Hammingova váha výstupní diference je 2 a maximální 6,259. Tato velmi malá čísla jsou důsledkem toho, že funkce MIX se skládá ze tří prvků XOR, ADD a ROT, přičemž při změně v nejvýznamnějším bitu nedochází u součtu ADD k šíření diference na sousední bity.



Obr. 9: Test lavinovitosti pro jednu rundu blokové šifry Threefish-512

### 3.1 Blízká pseudokolize u Skein-512

V kapitole 9.2 [4] je popsána blízká pseudokolize pro 8 rund kompresní funkce Skein-512. Tento útok využívá vlastnosti ADD popsané výše. Nechť δ označuje jednobitovou diferenci nejvýznamnějšího bitu 64 bitového slova δ = 1000…0. Jestliže pro Skein-512 máme diferenci klíče rovnu (0, 0, 0, 0, 0, 0, δ, δ) a diferenci tweaku rovnu (0, δ), pak dostáváme diferenci prvního subklíče před první rundou rovnu (0, 0, 0, 0, 0, 0, 0, δ), diferenci druhého subklíče po čtvrté rundě rovnu (0, 0, 0, 0, 0, 0, 0, 0) a diferenci třetího subklíče po osmé rundě rovnu (0, 0, 0, 0, δ, 0, 0, 0). V případě, že do vstupní zprávy zavedeme diferenci (0, 0, 0, 0, 0, 0, 0, δ), dostáváme po 8 rundách Threefish-512 diferenci s Hammingovou váhou = 1. Potom po použití módu řetězení dostaneme tomu odpovídající blízkou pseudokolizi s Hammingovou váhou 2, viz obr. 10. Toto platí pro libovolné hodnoty klíče, tweaku a zprávy. Výstupní diference po osmi rundách Threefish-512 odpovídá diferenci třetího subklíče (0, 0, 0, 0, δ, 0, 0, 0) a výstupní diference kompresní funkce Skein-512 po osmi rundách je (0, 0, 0, 0, δ, 0, 0, δ). Vliv této diference u algoritmu Threefish můžete sledovat na videu diference.mpg (49,7 MB, http://crypto-world.info/cw6/diference.mpg , [7] ).

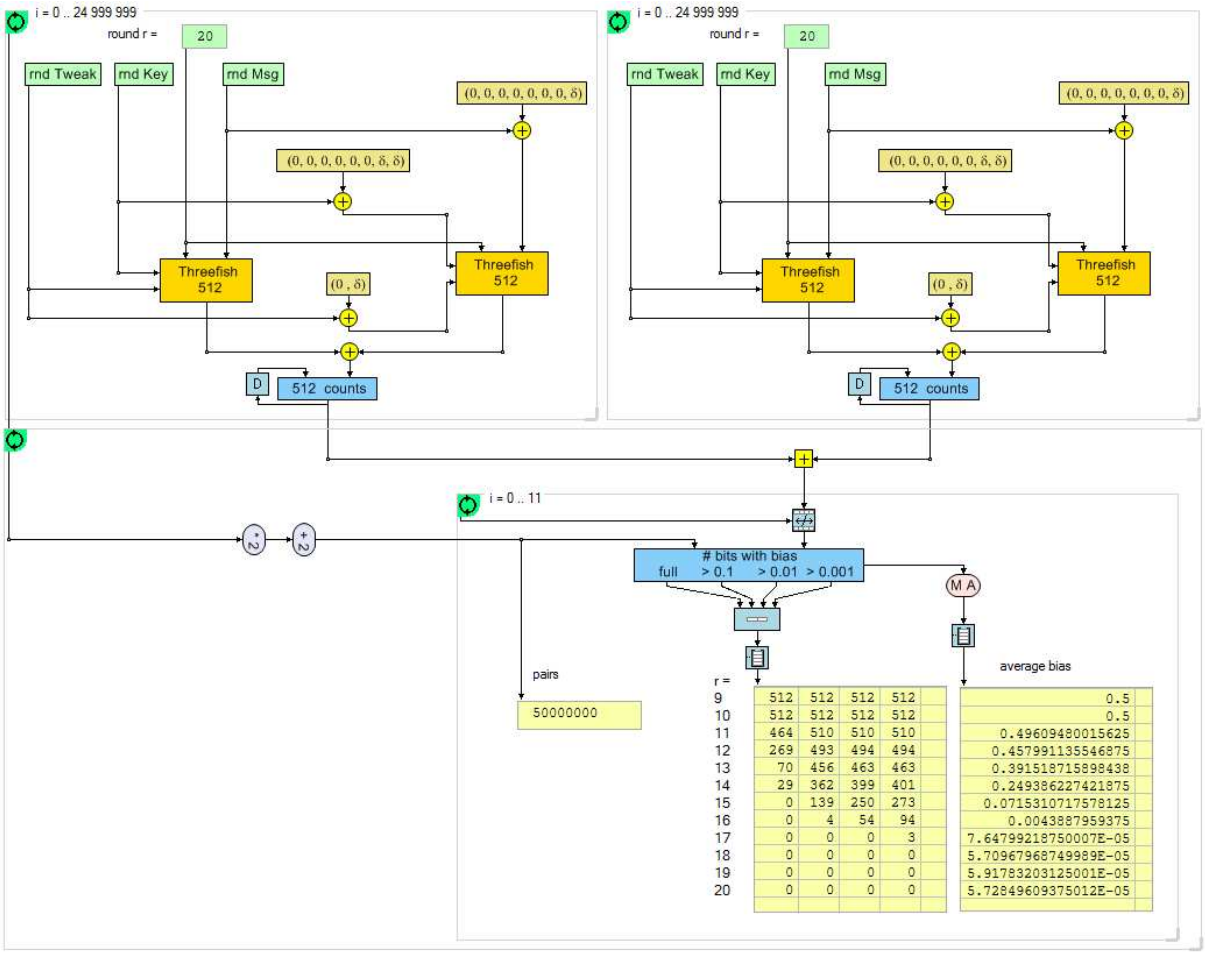



Obr. 10: Blízká pseudokolize pro osm rund Skein-512

Zajímalo nás, jak se tato zákonitost projevuje, když zvyšujeme počet rund z 8 na 20. Proto jsme v CipherCADu sestavili jednoduché schéma, jak je ukázáno na obr. 11. Schéma zachovává výše uvedené diference v klíčích, zprávách i tweacích, ale jejich hodnoty se generují náhodně (50,000,000 krát). Ukázali jsme, že pro deset rund je počet závislých bitů na výstupu roven 512, neboli výstupní diference platí přesně. Pro vyšší počet rund se toto

pravidlo začíná narušovat a počet bitů s přesnou reakcí ubývá a pravděpodobnost změny je stále bližší k hodnotě 0,5.

Obrázek 11 ukazuje počet závislých bitů (s významnou odchylkou) na výstupu 9. až 20. rundy Threefish-512. Celkový počet sledovaných vzorků je 50.000.000. Pokud máme dvoujádrový procesor, můžeme pustit dvě nezávislá schémata, každé pro 25,000,000 vzorků. Tím šetříme výpočetní čas. Po proběhnutí výpočtu v těchto dvou částech (iterátorech) můžeme pustit zbývající část v dolní části schématu. Ta vypočítává statistiky a prezentuje je v tabulce. Tabulka potvrzuje výsledky obdržené v [4].



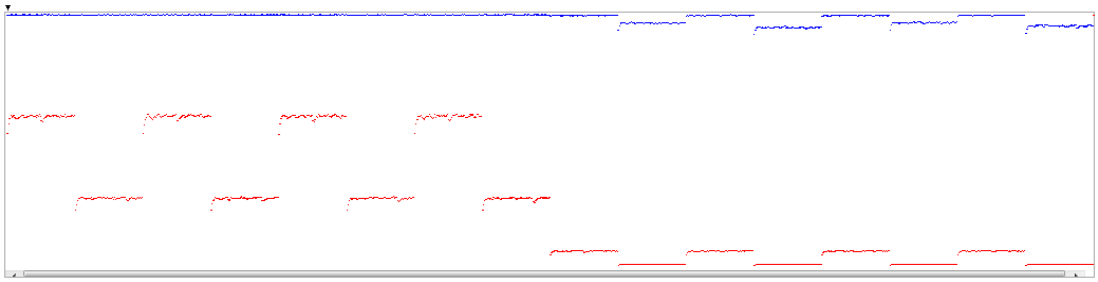

Obr. 11: Pozorování vlivu speciálních diferencí pro náhodné zprávy, klíče a tweaky v Threefish-512

### 3.2 Test lavinovitosti pro finalisty SHA-3

Abychom měli srovnání všech pěti finalistů SHA-3, ukážeme nyní výsledky stejného testu lavinovitosti i pro ostatní finalisty SHA-3. Jsou na obrázcích 12 - 15 a v tabulkách 1 - 4. V obrázcích jsme použili barevné odlišení: červená označuje výsledky pro jednu rundu, modrá dvě rundy, černá tři a zelená čtyři rundy.

Na obrázku 12 (13, 14, 15) metoda kráčejícího bitu ukáže diferenci, která má pro daný algoritmus nejmenší vliv na výstup kompresní funkce při malých hodnotách počtu rund, podobně jako u Skein-512.

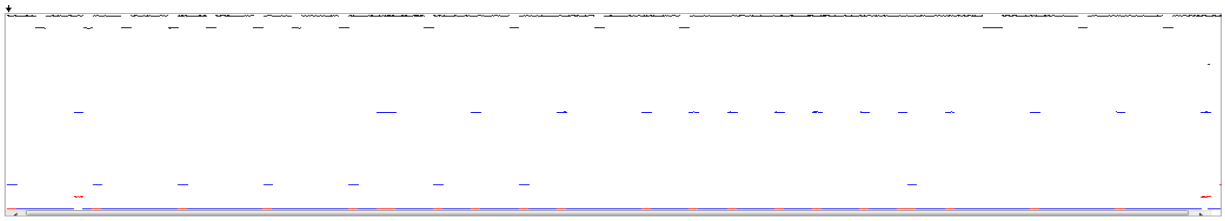
Takto vybraná diference byla použita v následujícím testu, jehož výsledky jsou v tabulkách 1 až 4.



Obr. 12: Metoda kráčejícího bitu 1. a 2. rundy funkce Rounds algoritmu BLAKE

| | # bits with bias | | | | average bias |
|---|---|---|---|---|---|
| r = | full | > 0.1 | > 0.01 | > 0.001 | |
| 1 | 920 | 1020 | 1020 | 1020 | 0.495117445195312 |
| 2 | 0 | 154 | 450 | 599 | 0.0420945248828125 |
| 3 | 0 | 0 | 0 | 0 | 5.77246093750001E-05 |

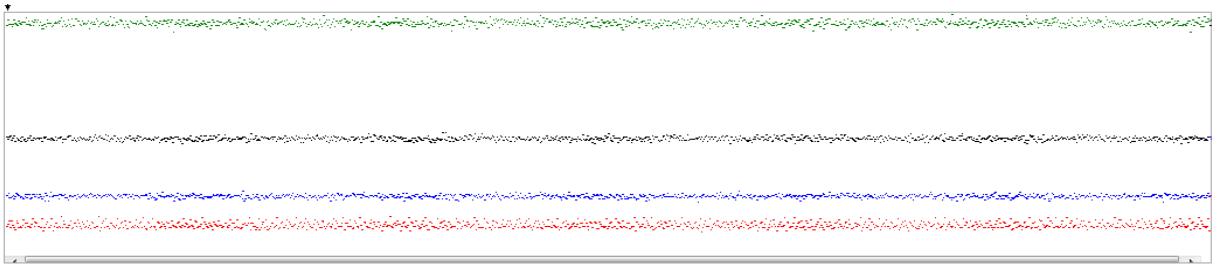
Tab. 1: Odchylka pro 50.000.000 náhodných vstupních párů funkce Rounds (BLAKE-512)



Obr. 13: Metoda kráčejícího bitu 1. až 3. rundy kompresní funkce f algoritmu Grøstl

| | # bits with bias | | | | average bias |
|---|---|---|---|---|---|
| r = | full | > 0.1 | > 0.01 | > 0.001 | |
| 1 | 1024 | 1024 | 1024 | 1024 | 0.5 |
| 2 | 1024 | 1024 | 1024 | 1024 | 0.5 |
| 3 | 64 | 64 | 64 | 64 | 0.0313080977734375 |
| 4 | 0 | 0 | 0 | 0 | 5.24451171874999E-05 |

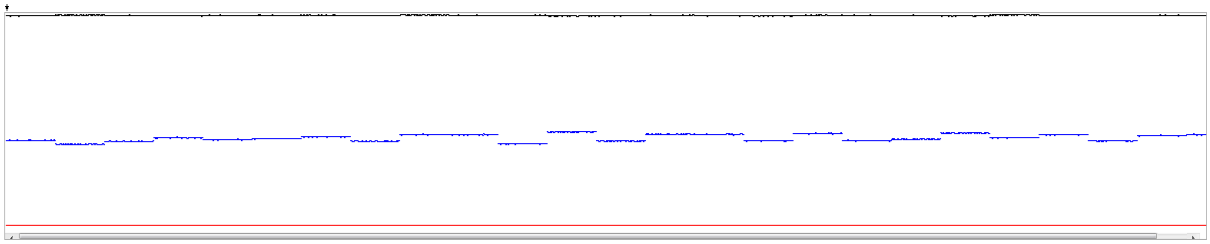
Tab. 2: Odchylka pro 50.000.000 náhodných vstupních párů kompresní funkce f (Grøstl-512)



Obr. 14: Metoda kráčejícího bitu 1. až 4. rundy funkce R8 algoritmu JH

| r = | # bits with bias full | > 0.1 | > 0.01 | > 0.001 | average bias |
|---|---|---|---|---|---|
| 1 | 1016 | 1017 | 1017 | 1017 | 0.496338434179688 |
| 2 | 1008 | 1008 | 1021 | 1021 | 0.492889431738281 |
| 3 | 992 | 992 | 1023 | 1024 | 0.485416377988281 |
| 4 | 960 | 960 | 1024 | 1024 | 0.470867342109375 |
| 5 | 896 | 896 | 1024 | 1024 | 0.441686526640625 |
| 6 | 768 | 768 | 1024 | 1024 | 0.383339388554688 |
| 7 | 512 | 512 | 1024 | 1024 | 0.266668562167969 |
| 8 | 0 | 0 | 1024 | 1024 | 0.0333348457226563 |
| 9 | 0 | 0 | 0 | 1024 | 0.0022223826171875 |
| 10 | 0 | 0 | 0 | 0 | 5.86367382812504E-05 |

Tab. 3: Odchylka pro 50.000.000 náhodných vstupních párů funkce R8 (JH-512)



Obr. 15: Metoda kráčejícího bitu 1. až 3. rundy funkce f[1600](A) algoritmu KECCAK

| r = | # bits with bias full | > 0.1 | > 0.01 | > 0.001 | average bias |
|---|---|---|---|---|---|
| 1 | 1578 | 1578 | 1578 | 1578 | 0.4931257669125 |
| 2 | 935 | 1259 | 1259 | 1259 | 0.3411845653 |
| 3 | 0 | 8 | 119 | 185 | 0.0037819483625 |
| 4 | 0 | 0 | 0 | 0 | 5.56877375000003E-05 |

Tab. 4: Odchylka pro 50.000.000 náhodných vstupních párů f[1600](A) (KECCAK-512)

Nyní máme výsledky pro všech pět algoritmů, ale potřebujeme je porovnat. Uděláme to jednoduše tak, že u každého algoritmu zjistíme z obdržených výsledků minimální počet rund, kdy už neexistují viditelné závislosti výstupních bitů na vstupním bitu.

Konkrétně hledáme minimální počet rund tak, aby počet bitů s odchylkou větší než 0.1% byl nulový. Výsledky jsou uvedeny souhrnně v tabulce 5. Protože počet rund je u všech algoritmů různý, je zde také vypočten poměr "počtu nebezpečných rund" k "celkovému počtu rund". Čím je tento poměr větší, tím je hašovací funkce konzervativnější.

I když jsme udělali jen jeden test, zkoumající vlastnost lavinovitosti, získali jsme určitou představu o bezpečnosti kandidátů SHA-3. Na základě uvedeného nemůžeme prohlásit, že Skein je nejbezpečnější z kandidátů, ale naopak vidíme, že rozdíly v naší míře bezpečnosti nejsou u jednotlivých kandidátů zásadní, viz tab. 5.

| algoritmus | testovaná funkce | celkový počet rund | min. počet rund | celk/min. | poznámka |
|---|---|---|---|---|---|
| BLAKE-512 | Rounds | 16 | 3 | 5,3 | z 14 na 16 rund<br>tweak pro 3 kolo SHA-3 |
| Grøstl-512 | f | 14 | 4 | 3,5 | |
| JH-512 | R8 | 42 | 10 | 4,2 | z 35.5 na 42 rund<br>tweak pro 3 kolo SHA-3 |
| KECCAK-512 | f[1600](A) | 24 | 4 | 6 | |
| Skein-512 | Threefish-512 | 72 | 18 | 4 | s pseudokolizí |
| | | | 9 | 8 | bez pseudokolize |

Tab. 5: Porovnání minimálního počtu rund splňujících test s odchylkou > 0, 1%

## 4 Závěr

V příspěvku jsme uvedli některé možnosti aplikace CipherCAD pro kryptoanalýzu. Ukazuje se, že je to velmi dobrý a názorný nástroj pro kryptoanalytické zkoumání i srovnávací analýzy. Ukázali jsme například intuitivní tvorbu modelu algoritmu Skein-512, testy lavinovitosti pro všech pět finalistů SHA-3 BLAKE, Grøstl, JH, KECCAK a Skein a srovnání jejich výsledků. Konkrétní závěr, který jsme v tomto jednom případě obdrželi je, že mezi uvedenými algoritmy nejsou žádné podstatné rozdíly.

## Literatura

[1] Sobotík, J., Plátěnka, V.: Aplikovaný výzkum a rozvoj pracoviště pro návrh a analýzu kryptografických systémů. Závěrečná zpráva k projektu NBÚ. Brno, 2009

[2] Domácí stránka týmu Skein: http://www.skein-hash.info/

[3] Stránka NIST k projektu SHA-3: http://csrc.nist.gov/groups/ST/hash/index.html

[4] Ferguson, N., Lucks, S., Schneier, B., Whiting, D., Bellare, M., Kohno, T., Callas, J., and Walker, J.: The Skein Hash Function Family. Version 1.3, 1 Oct 2010

[5] Klima, V., Platenka, V.: The Cryptographic Software Tool CipherCAD and Cryptanalysis. Security and Protection of Information 2011, May 10-12, 2011, Brno, Czech Republic, Proceedings of The Conference, ISBN 978-80-7231-777-6, pp. 54-65

[6] http://crypto-world.info/cw6/schema.mpg

[7] http://crypto-world.info/cw6/diference.mpg

# C. Rotorový šifrátor Fialka M-125
# Diel 3., Vybrané vlastnosti šifry

**Eugen Antal & Matúš Jókay, Kaivt FEI, STU v Bratislave**
( antal.87@gmail.com, matus.jokay@stuba.sk )

Rotorové šifrátory predstavujú zaujímavú oblasť skúmania klasických šifier. Hoci sa táto oblasť považuje za prekonanú, existujú stroje, ktoré predstavujú výzvu aj v súčasnosti.

Šifrátor Fialka M-125 môžeme považovať za jedno z vrcholných diel rotorových šifrovacích strojov. Patrí medzi najzáhadnejšie vo svojej kategórii a do dnešnej doby nie sú známe jeho kryptografické slabiny. Používala sa až do konca 90. rokov 20. storočia.

Rotorové šifrátory mali významné kryptografické nedostatky, o čom svedčí aj fakt, že mnohé z nich boli prelomené (Enigma, Hagelin, Siemens atď). Fialka M-125 predstavuje v tomto smere výnimku. Zatiaľ nie je známy úspešný útok na tento šifrátor. Aj vďaka tomu predstavuje v súčasnej kryptoanalýze zaujímavú oblasť skúmania.

Na kryptoanalýzu rotorových šifrátorov sa používajú rôzne matematické a štatistické vlastnosti zašifrovaného textu, ako aj charakteristiky mechanickej konštrukcie týchto šifrovacích strojov [3]. Pri (automatizovanej) kryptoanalýze šifrátorov sa využíva najmä frekvenčná charakteristika, náhodnosť rozdelenia ako aj zašifrovanie znakov na seba, perióda a odhad nastavenia rotorov [1][2]. Okrem matematického modelu šifrovacieho systému, ktorý realizuje ten-ktorý rotorový šifrátor, je dôležité pochopiť aj princíp fungovania týchto strojov ako elektromechanických zariadení.

V tomto článku, spracovanom podľa [4], sme sa zamerali na niektoré štatistické vlastnosti šifry Fialka M-125, ako aj jej modifikácií. Medzi najzaujímavejšie patrí analýza periódy zmenšenej verzie a bezpečnosť šifrovania vzhľadom na počet rotorov.

## 2 Zmenšená verzia Fialky

Vytvorenie zmenšenej verzie resp. modifikácia existujúceho algoritmu je veľmi jednoduchá. V našom prípade je potrebné modifikovať len niektoré vhodne zvolené časti algoritmu s dôrazom na zachovanie nasledovných mechanizmov:

- použitie pinov, ktorých polohy určujú zložitosť otáčania rotorov,
- cyklus dĺžky 3 na reflektore kvôli možnosti sebazašifrovania znaku a
- 2 nezávislé časti rotorov pre realizáciu dvoch smerov otáčania.

Dodržanie vymenovaných vlastností tvorí samotnú podstatu rotorového šifrátora Fialky.
Pre účely testovania sme zvolili verziu so 4 rotormi a s abecedou Z6. Táto zjednodušená verzia je najmenšia možná. Zvolený počet rotorov nemôže byť menší, pretože by nebolo možné zachovať zároveň oba smery otáčania a význam blokovacích pinov. Abeceda je zvolená podľa minimálnych kritérií sebazašifrovania znaku na reflektore (R(u)). Jedno písmeno (kontakt) vracia vstupný znak ako výstup. Keď sa na reflektore odrazí ten istý signál, na ktorý aplikujeme inverzné operácie, výsledkom je vstupujúci znak.

Vieme, že je potrebný jeden cyklus dĺžky 3, aby sme zabezpečili párny počet kontaktov s korektným prepojením. Preto pridáme tri kontakty a na záver sa pridá ešte jeden štandardný cyklus dĺžky 2 (pridáme ešte ďalšie dva kontakty).

Po týchto úpravách môžeme proces šifrovania symbolicky zapísať nasledujúcou rovnicou:

$$y = S^{-1}\rho_4^{-1}\rho_3^{-1}\rho_2^{-1}\rho_1^{-1}R(u)\rho_1\rho_2\rho_3\rho_4 S\ (x)$$

kde S je vstupná substitúcia, R je substitúcia na reflektore, x vstupný znak z klávesnice, y výstupný zašifrovaný znak. Ostatné symboly sú použité pre permutácie rotorov.

# 3 Rozšírená verzia Fialky na šifrovanie ľubovoľného znaku

Pôvodná verzia Fialky umožňuje šifrovanie abecedy Z30, na ktorú je možné použiť ľubovoľné mapovanie znakov. Nedovoľuje však šifrovať ľubovoľné typy údajov (napríklad obrázky, ktoré používajú vstupnú abecedu s rozsahom 256 znakov). Na šifrovanie takýchto dát by bolo treba rozšíriť abecedu Fialky na Z256, čo je však nevýhodné z pohľadu zložitosti výpočtov a veľkosti potrebných dát (napr. pre substitučné tabuľky).

Preto sme zvolili iný spôsob úpravy – zmenu mapovania abecedy otvoreného textu (OT). Abeceda OT sa skladá z 256 symbolov, ktoré je možné v počítači reprezentovať jedným bajtom (8 bitov). Nasledujúci algoritmus popisuje nami navrhnutú zmenu kódovania OT:

1. bitová reprezentácia znaku x otvoreného textu sa rozdelí na 2 polovice: xH a xL,
2. spodná polovica xL (reprezentovaná 4 bitmi) je v rozsahu 0 až 15,
3. horná polovica xH(taktiež reprezentovaná 4 bitmi) sa transformuje tiež do rozsahu 0 až 15, a to príslušným bitovým posunom.

Vstupný znak je po aplikovaní tohto postupu rozdelený na dve časti v potrebnom rozsahu. Teraz už dokážeme šifrovať upravený vstup s pôvodným algoritmom bez úprav abecedy pôvodného šifrátora. Po zašifrovaní OT je potrebné vyriešiť ešte ďalší problém. Zašifrovaný text je v rozsahu Z30, teda jeho bitová reprezentácia využíva 5 bitov. Pri spájaní dvoch nezávisle zašifrovaných častí jedného vstupného bajtu OT nám vznikne údaj s veľkosťou 10 bitov (každá štvorica bitov je šifrovaním mapovaná na päticu). To znamená, že v prípade, keď je hodnota zašifrovaného symbolu väčšia ako 15, sa orezaním pätice výstupných bitov na štvoricu jeden informačný bit stratí. Táto strata (maximálne 2 bitov) spôsobí, že zašifrovaný text nie je jednoznačne dešifrovateľný.

Aby sme sa vyhli komplikáciám bitovej straty (spôsobenej prekódovaním 5 bitov na 4 bity a opačne), navrhli sme 3 možné spôsoby šifrovania osembitových symbolov:

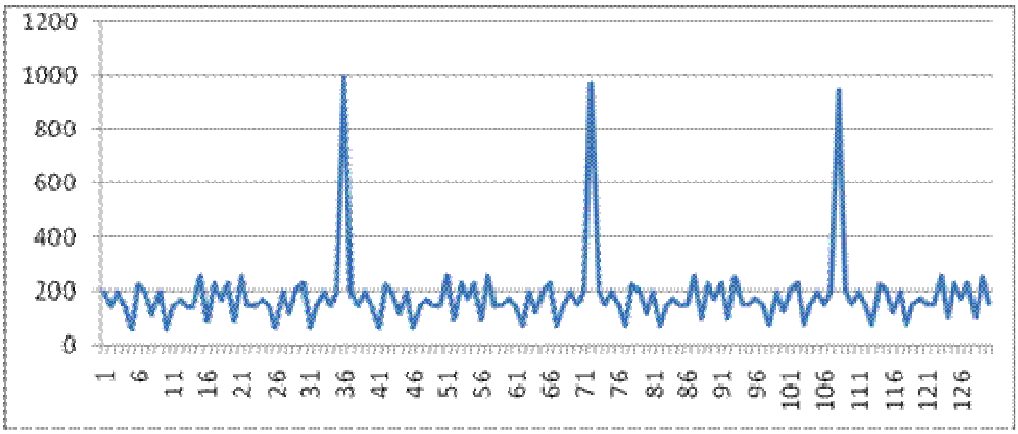
- zmenšiť abecedu šifrátora na Z16, takže 4 bity vstupu sa zobrazia na 4 bity výstupu,
- zväčšiť abecedu šifrátora na Z256, aby bolo možné šifrovať naraz celý bajt, alebo
- opakovane šifrovať znak dovtedy, kým sa na výstupe neobjaví 4-bitová hodnota.

Podrobnejšie o bezpečnosti jednotlivých spôsobov šifrovania je v kapitole 5.

# 4 Perióda

Na zistenie periódy sme navrhli nasledovné testy založené na autokorelácii. Algoritmus je tento:

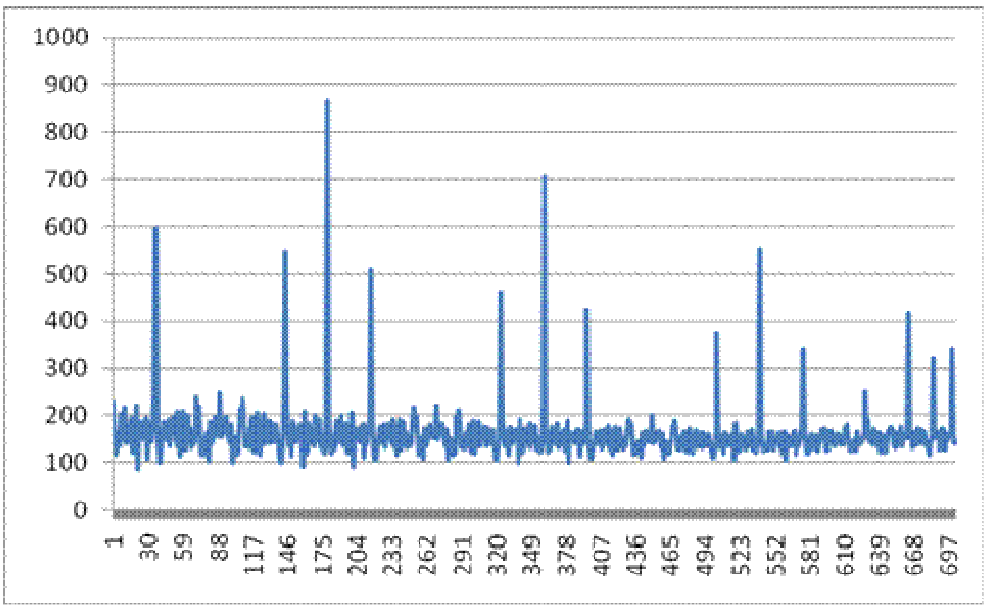
1. Zoberieme **n** (veľkosť abecedy zašifrovaného textu) fixných otvorených textov dĺžky 1024 znakov. Prvky i-teho zašifrovaného textu sú vždy rovnaké (samé 0, samé 1, ... , až napokon samé hodnoty n-1).
2. Vyberieme náhodné kľúče.
3. Zašifrujeme príslušný otvorený text.
4. Získaný zašifrovaný text (ZT) posúvame o jeden znak doprava a porovnávame zhodujúce sa prvky s predchádzajúcim posunutým ZT.



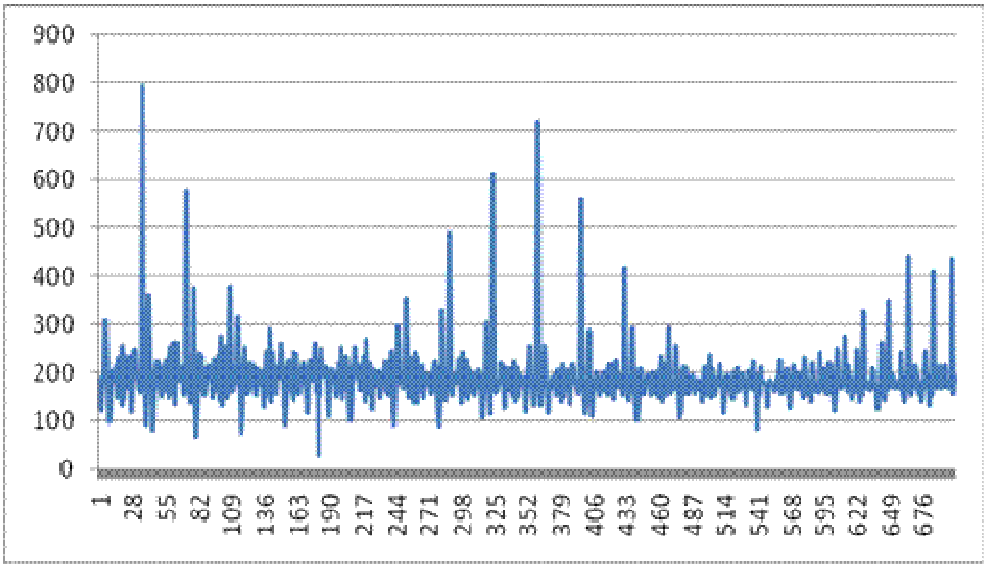

**Obr. 1.** Autokorelačná funkcia pre OT samé 0.

V zmenšenej verzii Fialky sme v tomto teste pre každý otestovaný OT dostali periódu 36 (viď. obr. 1). Zvolená štruktúra OT zaručuje optimálny spôsob určenia periódy.

V ďalších testoch sme sa pokúsili určiť, či pri miernej modifikácií otvoreného textu dostaneme takú istú resp. podobne malú periódu. Použili sme ten istý algoritmus ako v predošlom teste, odlišnosti boli v inicializácii otvoreného textu. Pre tento test sme stanovili pravidlo, že sa znaky OT striedajú po 5, 10, 20 a 50 opakovaniach.



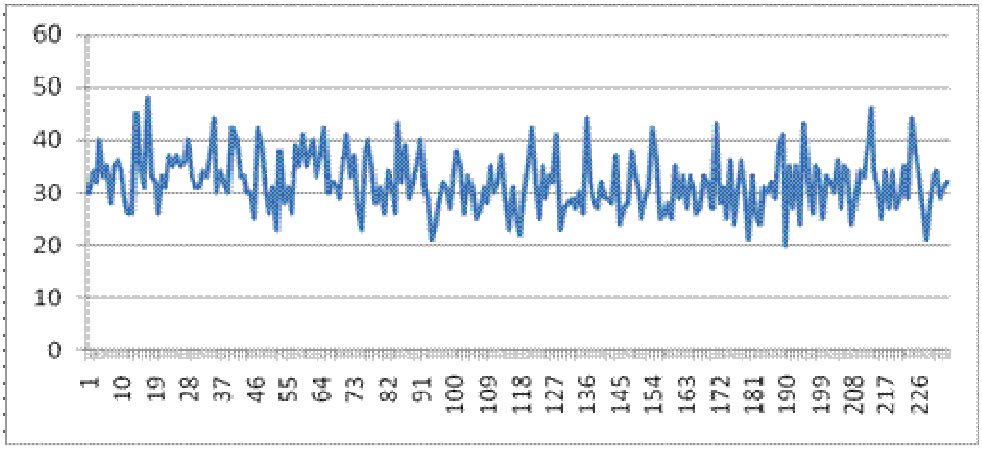

**Obr. 2.** Autokorelačná funkcia pre OT striedania symbolov 0 a 1 po 10 opakovaniach.




**Obr. 3.** Autokorelačná funkcia pre OT striedania symbokov 0 a 1 po 20 opakovaniach.

Po vyhodnotení výsledkov bolo otázne, či táto nízka perióda je len špecifickou vlastnosťou pre danú konkrétnu permutáciu rotorov a umiestnenie pinov, alebo všeobecne platí aj pre všetky ostatné rotorové permutácie. Preto sme predošlé testy opakovali aj pri iných nastaveniach šifrátora (rôzne permutácie a umiestnenia blokovacích pinov). Výsledky mali podobný charakter ako v predošlých testoch.

Na základe uskutočnených testov sme sa rozhodli realizovať aj autokorelačný test (šifrovanie otvoreného textu zloženého z rovnakých symbolov) aj na plnej verzii Fialky. Tým sme chceli získať porovnanie so zmenšenou verziou. Testy na plnej verzii šifrátora priniesli úplne iné výsledky a z nameraných hodnôt sa nedala určiť žiadna (nízka) perióda (viď obrázok č. 4).




**Obr. 4.** Autokorelačná funkcia plnej verzie šifrátora pre OT samé 0.

# 5 Bezpečnosť založená na počte rotorov

V kapitole 3 sme navrhli niekoľko spôsobov modifikácie Fialky, ktorá by umožňovala šifrovať ľubovoľné znaky v osembitovej reprezentácii. Overiť kvalitu takto modifikovaného šifrátora je jednoduché – vizuálne skontrolovať výsledok šifrovania obrázku obsahujúceho monotónne farebné plochy (t.j. časti rovnakej farby).

Táto metóda je veľmi efektívna predovšetkým pri analýze blokových šifier. Blokové šifry majú jeden veľký nedostatok – pri šifrovaní rovnakého otvoreného textu tým istým kľúčom dostávame rovnaký výstup. Toto umožňuje identifikovať rovnaké bloky. Použitím štatistických metód aplikovaných na štruktúru zašifrovaného textu je možné v istých prípadoch získať informácie uľahčujúce kryptoanalýzu. Preto je potrebné v prípade blokových šifier používať dodatočné módy spätnej väzby medzi jednotlivými blokmi.

V našom prípade je však situácia iná. Šifra Fialka je prúdová s meniacim sa kľúčom pre každý nový vstup. Pri šifrovaní dostatočne veľkého vstupu je však možné analyzovať periódu. Keď zoberieme dostatočne veľký obrazový súbor s monotónnymi plochami, môžeme analyzovať bezpečnosť šifry vzhľadom na premenlivý počet rotorov.

Obrazový test má nasledovnú štruktúru:

1. Definovanie pokusného obrázku s veľkými monotónnymi plochami.
2. Zašifrovanie daného obrázku pomocou 10, 8, 6, 4 a 2 rotorov.
3. Dešifrovanie a vizuálna kontrola takto získaných obrázkov.

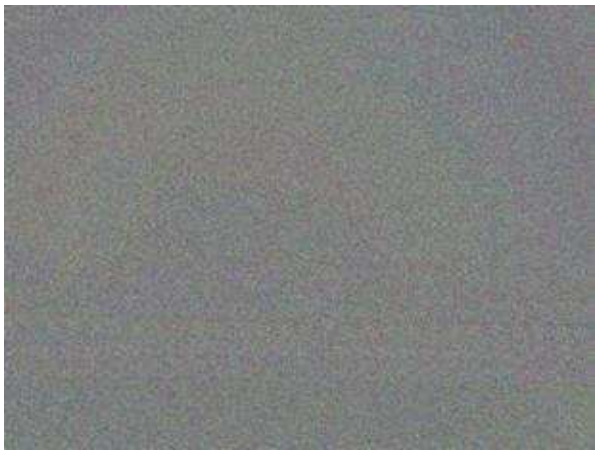
**Obr. 5.** Testovací obrázok.

Označenie jednotlivých spôsobov šifrovania osembitových symbolov (opísané v kapitole 3) je:

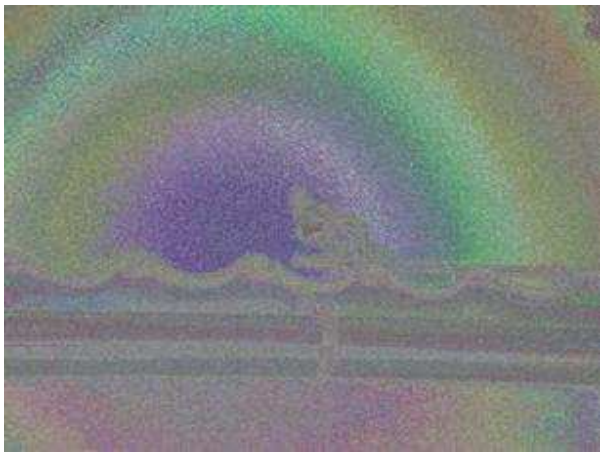
- Z30_T_R – pôvodná abeceda šifrátora (Z30), opakované šifrovanie (až kým nie je na výstupe znak reprezentovateľný 4 bitmi), transformácia symbolu otvoreného textu (8 bitov) na 2 časti (každá má veľkosť 4 bitov),
- Z16_T – zredukovaná abeceda šifrátora na 16 znakov, transformácia symbolu otvoreného textu (8 bitov) na 2 časti (každá má veľkosť 4 bitov),
- Z256 – rozšírená abeceda šifrátora na 256 znakov (nie je nutná transformácia symbolov otvoreného textu).

Výsledky šifrovania pre algoritmus Z30_T_R (obrázky č. 6 a 7) preukázali väčšiu bezpečnosť, než sme predpokladali. Pri použití 6 rotorov nie sú viditeľné ešte žiadne črty

pôvodného obrázku. Až pri 4 rotoroch sa objavujú prvé pozorovateľné artefakty. Jedine pri redukcii na najnižší možný počet rotorov (na 2) je možné považovať túto verziu za zraniteľnú.



**Obr. 6.** Z30_T_R po zašifrovaní 4 rotormi.



**Obr. 7.** Z30_T_R po zašifrovaní 2 rotormi.

V prípade Z_16_T (obrázky č. 8, 9 a 10) je pri 6 rotoroch výsledok menej náhodný (oproti Z30_T_R) – je tam viditeľná istá perióda, ale nie je možné identifikovať obsah obrázku. Pri použití 4 rotorov je pozorovateľná štruktúra pôvodného obrázku a pri 2 rotoroch je možné obsah obrázku jednoznačne identifikovať.

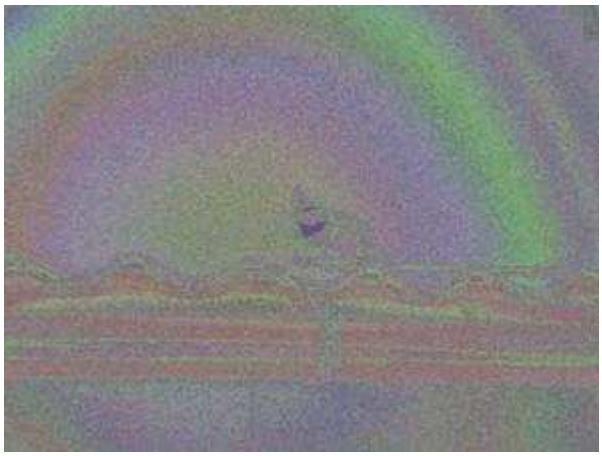
**Obr. 8.** Z16_T po zašifrovaní 6 rotormi.

**Obr. 9.** Z16_T po zašifrovaní 4 rotormi.

**Obr. 10**. Z16_T po zašifrovaní 2 rotormi.

Vizuálnou kontrolou môžeme ako najbezpečnejšiu ohodnotiť verziu Z_256 (viď obrázky č. 11 a 12). Podobne ako pri Z30_T_R, aj v tomto prípade je možné identifikovať črty pôvodného obrázku len pri 2 rotoroch. Avšak použitie minimálneho počtu rotorov prezrádza najmenej informácií v porovnaní so všetkými predošlými testami.

**Obr. 11.** Z256 po zašifrovaní 4 rotormi.



**Obr. 12.** Z256 po zašifrovaní 2 rotormi.

Na záver môžeme konštatovať, že všetky otestované verzie sú bezpečné (vzhľadom na vizuálnu kontrolu) pri použití 8 rotorov. Pri redukcii na 2 (resp. 4) rotory nie je možné považovať ani jednu variantu tejto šifry za bezpečnú. Experimentálne bolo poukázané, že aj keď je sila šifry vo väčšej miere závislá od počtu rotorov, dôležitá je aj voľba veľkosti použitej abecedy.

# Literatúra

[1] O. GROŠEK – P. ZAJAC: Automated cryptanalysis of classical ciphers. In: J. R. Rabunal Dopico, J. Dorado De La Calle, A. Pazos Sierra (Eds.): Encyclopedia of Artificial Intelligence, 2009, ISBN: 978-1-59904-849-9. pp. 186-191.

[2] O. GROŠEK – P. ZAJAC: Automated cryptanalysis. In: J. R. Rabunal Dopico, J. Dorado De La Calle, A. Pazos Sierra (Eds.): Encyclopedia of Artificial Intelligence, 2009, ISBN: 978-1-59904-849-9. pp. 179-185.

[3] O. GROŠEK – M. VOJVODA – P. ZAJAC: Klasické šifry. - Bratislava : STU v Bratislave, 2007. - 214 p. - ISBN 978-80-227-2653-5.

[4] E. ANTAL: Niektoré problémy kryptoanlýzy šifry Fialka M-125. Diplomová práca, FEI STU Bratislava, 2011.

# D. KEYMAKER – studentská soutěž
## v rámci workshopu Mikulášská kryptobesídka
## 1.–2. prosinec 2011, Praha , http://mkb.buslab.org/

Mikulášská kryptobesídka přichází letos již v 11. ročníku. Je zaměřena na podporu úzké spolupráce odborníků se zájmem o teoretickou a aplikovanou kryptografii a další příbuzné oblasti informační bezpečnosti. Hlavním cílem je vytvořit prostředí pro neformální výměnu informací a nápadů z minulých, současných i budoucích projektů. Cítíme potřebu setkání expertů s jejich kolegy bez obchodních vlivů, starostí s (potenciálními) zákazníky, šéfy a dalšími rozptylujícími faktory. ;-)

Workshop se skládá ze dne prezentací příspěvků, diskusí a neformálního setkání ve čtvrtek *1. prosince 2011* a půldne prezentací příspěvků a diskusí v pátek *2. prosince 2011*. Pro workshop jsou domluveny zvané příspěvky:

- Chris Mitchell (Royal Holloway, UK): *New architectures for identity management - removing barriers to adoption*.
- Graham Steel (INRIA, Francie): *Attacking and Fixing PKCS#11 Security Tokens*.
- Viktor Fischer (Jean Monnet University Saint-Etienne, Francie): *Recent Advances in Random Numbers Generation for Cryptography*.
- Pavel Vondruška (Telefónica O2 Czech Republic): *Šifry používané československými osobnostmi*.
- Jozef Kollár (SvF STU v Bratislave, SR): *Československé šifry z obdobia 2. svetovej vojny*.

### KEYMAKER – Pokyny pro autory

Přijímány jsou příspěvky zaměřené především na oblasti kryptoanalýzy, aplikované kryptografie, bezpečnostních aplikací kryptografie, počítačové a komunikační bezpečnosti a dalších souvisejících oblastí. Návrhy se přijímají odděleně pro sekci KEYMAKER (studentská soutěž) a pro hlavní program workshopu. Příspěvek pro KEYMAKER má požadovaný rozsah 5-15 stran A4 a připravenost pro anonymní hodnocení (bez jmen autorů a zjevných odkazů). Identifikační a kontaktní údaje prosím pošlete v těle e-mailu s příspěvkem jakožto přílohou a jasným označením KEYMAKER. Přijímány jsou články, bakalářské či diplomové práce, nebo jiná kvalitní ucelená díla, kde v případě rozsahu nad 15 stran požadujeme výtah podstatného obsahu v max. rozsahu 8 stran, s vlastní prací jako přílohou.

Mezi autory nejlepších příspěvků PV rozdělí *finanční odměny v celkové výši 105 tisíc Kč*. Oceněno bude min. 3 a max. 7 příspěvků. Návrhy příspěvků budou posouzeny PV a autoři budou informováni o přijetí/odmítnutí do *31. října 2011*. Příspěvek pak musí být prezentován na workshopu.

Šablony pro formátování příspěvků pro Word a LaTeX lze získat na www stránkách workshopu: http://mkb.buslab.org. Příspěvky mohou být napsané v češtině, slovenštině, nebo angličtině.

Příspěvky připravené podle výše uvedených pokynů zasílejte ve formátu PDF, příp. RTF a to tak, aby na uvedenou adresu přišly nejpozději do *3. října 2011*. Pro podávání příspěvků prosím použijte adresu matyas ZAVINAC fi.muni.cz a do předmětu zprávy uveďte „MKB 2011 – navrh prispevku KEYMAKER“. Příjem návrhů bude potvrzován do dvou pracovních dnů od přijetí.

### Důležité termíny

| | |
|---|---|
| Návrhy příspěvků: | 3. října 2011 |
| Oznámení o přijetí/odmítnutí: | 31. října 2011 |
| Konání MKB 2010: | 1. – 2. prosince 2011 |

### Programový výbor

Dan Cvrček, Smart Architects, UK
Martin Drahanský, VUT v Brně, ČR
Petr Hanáček, VUT v Brně, ČR
Vlastimil Klíma, KNZ, ČR
Vašek Matyáš, FI MU, Brno, ČR – předseda
Tomáš Rosa, Raiffeisenbank a UK, ČR
Luděk Smolík, Siegen, SRN
Martin Stanek, UK, Bratislava, SR
Petr Švenda, FI MU, Brno, ČR
Petr Švenda, FI MU, Brno, ČR

# E. Konference EUROPEN 2011

**Výzva k podávání příspěvků (Call for papers)**
Konference EUROPEN 2011 (http://www.europen.cz)
2-5. října 2011, Areál kláštera Želivy u Humpolce

Podzimní setkání uživatelů otevřených systémů EUROPEN 2011 proběhne se *zaměřením na praktickou bezpečnost počítačových systémů a aplikovanou kryptografii*. Cílem setkání je umožnit výměnu nových myšlenek a nápadů, prezentaci výsledků aktuálních projektů a zkušenosti s použitím nových postupů relevantních z hlediska bezpečnosti.

Přijímány jsou příspěvky zaměřené především na oblast bezpečnosti mobilních zařízení, využití kryptografických čipových karet, síťové bezpečnosti a aplikované kryptografie. Lze ale zasílat i relevantní příspěvky mimo tyto oblasti. Výhodou je důraz na využití otevřených systémů a praktické zkušenosti s implementací.

Své návrhy (rozšířený abstrakt, 1-2 strany) zasílejte připravené pro anonymní hodnocení (bez jmen autorů a zjevných odkazů) na adresu svenda zavinac fi.muni.cz do *13.června 2011* s označením EUROPEN2011 včetně kontaktních údajů autorů. Doporučený rozsah finálního příspěvku je 2-10 stránek, není však striktně omezen. Příspěvky budou později sázeny v TeXu, jeho využití ale není pro návrhy příspěvků požadováno.

Přijímány budou příspěvky v anglickém, českém a slovenském jazyce. Návrhy příspěvků budou posouzeny programovým výborem a autoři budou o přijetí/odmítnutí informováni do *27. června 2011*. U přijatých příspěvků bude vyžadována krátká anotace (jeden odstavec) a CV autorů pro propagační brožuru.

**Důležité termíny:**
- Podání návrhu příspěvku (rozšířený abstrakt): 13. června 2011
- Oznámení o přijetí/odmítnutí: 27. června 2011
- Odevzdání krátké anotace a CV pro propagační materiály: 18. července 2011
- Odevzdání finálního příspěvku pro sborník: 5. září 2011
- Konání konference: 2 - 5. října 2011

**Programový výbor:**
Vašek Matyáš (předseda, Masarykova univerzita)
Marek Kumpošt (Masarykova univerzita, Trusted Network Solutions a.s.)
Marián Novotný (ESET spol. s. r.o.)
Josef Pojsl (Trusted Network Solutions a.s.)
Zdeněk Říha (Masarykova univerzita)
Roman Štěpánek (SODATSW spol. s r.o.)
Petr Švenda (Masarykova univerzita)

# F. O čem jsme psali v červnu 2000 – 2010

**Crypto-World 6/2000**



**Crypto-World 6/2001**



**Crypto-World 6/2002**



**Crypto-World 6/2003**



**Crypto-World 6/2004**



**Crypto-World 6/2005**

**Crypto-World 6/2006**



**Crypto-World 6/2007**


Příloha: Mikulášská kryptobesídka (6.-7.12.2007)- MKB2007_CallForPapers_cerven.pdf

**Crypto-World 6/2008**



**Crypto-World 6/2009**


Příloha: javascript-priloha.pdf (179 kB)
javascript-priloha_1_3.rtf (64 kB)

**Crypto-World 6/2009**

A. Výprava za obsahem javascriptu (J.Vorlíček, J.Suchý) 2-6
B. Anonymita v globální síti (J.Hajný) 7-11
C. Formát elektronické fakturace ISDOC (P.Kuchař) 12-18
D. Malá soutěž v luštění RSA (P.Vondruška) 19-20
E. O čem jsme psali v červnu 1999-2008 21-22
F. Závěrečné informace 23
Příloha: javascript-priloha.pdf (179 kB)
javascript-priloha_1_3.rtf (64 kB)

**Crypto-World 6/2010**

# G. Závěrečné informace

### 1. Sešit

Crypto-World je oficiální informační sešit "Kryptologické sekce Jednoty českých matematiků a fyziků" (GCUCMP). Obsahuje články podepsané autory. Případné chyby a nepřesnosti jsou dílem autorů jednotlivých podepsaných článků, GCUCMP za ně nemá odbornou ani jinou zodpovědnost.
Adresa URL, na níž můžete najít tento sešit (zpravidla 3 týdny po jeho rozeslání) a předchozí e-ziny, denně aktualizované novinky z kryptologie a informační bezpečnosti, normy, standardy, stránky některých členů a další související materiály: http://crypto-world.info



### 2. Registrace / zrušení registrace

Zájemci o e-zin se mohou zaregistrovat pomocí e-mailu na adrese pavel.vondruska@crypto-world.info (předmět: Crypto-World) nebo použít k odeslání žádosti o registraci elektronický formulář na http://crypto-world.info. Při registraci vyžadujeme pouze jméno a příjmení, titul, pracoviště (není podmínkou) a e-mail adresu určenou k zasílání kódů ke stažení sešitu.
Ke zrušení registrace stačí zaslat krátkou zprávu na e-mail pavel.vondruska@crypto-world.info (předmět: ruším odběr Crypto-Worldu!) nebo opět použít formulář na http://crypto-world.info. Ve zprávě prosím uveďte jméno a příjmení a e-mail adresu, na kterou byly kódy zasílány.

### 3. Redakce

**E-zin Crypto-World**

| | |
|---|---|
| Redakční práce: | Pavel Vondruška<br>Vlastimil Klíma<br>Tomáš Rosa<br>Dušan Drábik |
| Přehled autorů: | http://crypto-world.info/obsah/autori.pdf |
| NEWS | Jaroslav Pinkava |
| Webmaster | Pavel Vondruška, jr. |

### 4. Spojení (abecedně)

| | | |
|---|---|---|
| redakce e-zinu | ezin@crypto-world.info , | http://crypto-world.info |
| Vlastimil Klíma | v.klima@volny.cz , | http://cryptography.hyperlink.cz/ |
| Jaroslav Pinkava | jaroslav.pinkava@gmail.com , | http://crypto-world.info/pinkava/ |
| Tomáš Rosa | tomas.rosa@rb.cz , | http://crypto.hyperlink.cz/ |
| Dušan Drábik | Dusan.Drabik@o2bs.com , | |
| Pavel Vondruška | pavel.vondruska@crypto-world.info | http://crypto-world.info/vondruska/index.php |
| Pavel Vondruška, jr. | pavel@crypto-world.info , | http://webdesign.crypto-world.info |