Crypto-World

Informační sešit GCUCMP

ISSN 1801-2140

Ročník 16, číslo 1-3/2014 **9. března**

1-3/2014

Připravil: Mgr. Pavel Vondruška

Sešit je přednostně distribuován registrovaným čtenářům.
Starší sešity jsou dostupné na adrese
http://crypto-world.info/
(1380 registrovaných odběratelů)

Obsah :

A. Československá šifra TTS a jej lúštenie

Peter Javorka, pjavor1338@gmail.com

ÚIM, FEI STU v Bratislave

Abstrakt

Tu popisovaná šifra TTS sa používala napr. počas 2. svetovej vojny československou exilovou vládou v Londýne, pre spojenie s domácim odbojom, Moskvou a Istanbulom. Je založená na dvojitej transpozícii a následnej substitúcii znakov. [1]

1 Šifrovanie TTS

Pre substitúciu znakov sa používala napr. nasledovná substitučná tabuľka, ktorá sa posúvala podľa dňa šifrovania. Tzn. prvý deň mesiaca sa A šifrovalo ako 01, druhý deň 02 atď. K šifrovaniu sa používala neúplná česká abeceda s číslami a niektorými špeciálnymi znakmi.

	0	1	2	3	4	5	6	7	8	9
0		A	B	C	Č	D	E	Ě	F	G
1	H	I	J	K	L	M	N	O	P	Q
2	R	Ř	S	Š	T	U	V	W	X	Y
3	Z	Ž	.	?	-	/	1	2	3	4
4	5	6	7	8	9	0				

Heslá transpozičných tabuliek sa vyberali ako text na daných riadkoch z vopred dohodnutých kníh[1].

Rôzne zdroje uvádzajú rôznu dĺžku hesiel. V [1] sa uvádza aspoň 15 znakov, podľa [2] je to 12-17 znakov. V mojej implementácii šifrovacieho programu som sa rozhodol použiť heslá s dĺžkou 15-23 znakov.

Pri výbere hesla z danej knihy sa tiež zohľadňoval dátum. Pre každý mesiac bola určená strana, z ktorej sa heslá vyberali. Jednotlivé dni potom mali určenú dĺžku oboch hesiel a riadok z danej strany. Vznikol tak zoznam trojíc, kde prvé číslo znamená deň šifrovania, druhé dĺžku prvého transpozičného hesla a tretie dĺžku druhého transpozičného hesla (napr. 1-16-20, 2-17-15 ...).

Po príprave substitučnej tabuľky a transpozičných hesiel pre daný deň, bolo potrebné text pripravený na šifrovanie rozdeliť na menšie časti. Delil sa približne po 50 znakov tak, aby na konci časti bolo vždy celé slovo. Druhým krokom bolo označiť jednotlivé časti správy. Na začiatok a koniec sa pridalo „/“ a písmeno danej časti. Výnimkami boli prvá časť, v ktorej sa neoznačoval začiatok, a posledná časť, kde sa neoznačoval koniec.

Predstavme si teda prípad, že chceme zašifrovať text, ktorý je dlhý 170 znakov. Rozdelíme si ho na časti dlhé 50, 55 a 65 znakov. Na koniec prvej časti pridáme znaky „/“. Na začiatok druhej pripíšeme „A/“ a na koniec „B“ a tretia časť bude začínať „B/“. Je veľmi podstatné aby jednotlivé časti nemali rovnakú dĺžku,

[1] napr. T. G. Masaryk: *Světová revoluce* alebo Fabricius: *Levi hladujú v Neapole* [2]

pretože pokiaľ budeme šifrovať časti rovnakej dĺžky v jeden deň (tzn. s rovnakými dĺžkami hesiel transpozičných tabuliek), výrazne tým uľahčíme prácu prípadnému lúštiteľovi. Toto pravidlo sa počas vojny často porušovalo a nemeckí lúštitelia boli schopní depeše šifrované týmto spôsobom bez problémov lúštiť.

Ukážme si šifrovanie prakticky[2]. Použijeme stranu 188 z prvej časti Tolkienovho Pána prsteňov pre vytvorenie hesiel. Povedzme, že šifrujeme tretí deň v mesiaci. Z tretieho riadku vyberieme prvých 16 znakov zo začiatku a 17 od konca. Vzniknú tieto heslá: **VNUTILIFRODOVIDA**, **SELSPUSTITESTERAZ**. Potrebujeme ich číselné vyjadrenie, takže si urobíme tabuľku a jednotlivé písmená ohodnotíme tak ako nasledujú v abecede.

V	N	U	T	I	L	I	F	R	O	D	O	V	I	D	A
15	9	14	13	5	8	6	4	12	10	2	11	16	7	3	1

S	E	L	S	P	U	S	T	I	T	E	S	T	E	R	A	Z
9	2	6	10	7	16	11	13	5	14	3	12	15	4	8	1	17

Ako text použijeme úrivok z diela *Keď báčik z chocholova umrie* od Martina Kukučína.

„Ja som nie opilý. Zdržím veľmi moc. U báčika popijem toľko moc!“
„A kto je to ten báčik?“
„Dekan v Chocholove.“
„Ten? Toho znám,“odpovedal Ondrej naraz vážne.
Adušovo zdelenie urobilo naň veľký dojem.
„Znáš ho? Ako?“
„Kúpil som od neho tejto jesen...“

V texte zameníme, prípadne odstránime znaky, ktoré sa nenachádzajú v substitučnej tabuľke. Medzery nahradíme pomlčkami a rozdelíme text na časti. Nesmieme zabudnúť označiť nadväznosť:

JA-SOM-NIE-OPILY.ZDRŽIM-VELMI-MOC.U-BAČIKA-POPIJEM-TOLKO-MOC.
A-KTO-JE-TO-TEN/A

A/BAČIK?DEKAN-V-CHOCHOLOVE.TEN?TOHO-ZNAM.
ODPOVEDAL-ONDREJ-NARAZ-VAŽNE.ADUŠOV/B

B/O-ZDELENIE-UROBILO-NAN-VELKY-DOJEM.ZNAŠ-HO?
AKO?KUPIL-SOM-OD-NEHO-TEJTO-JESEN

Takto pripravené správy transponujeme podľa prvej a následne podľa druhej tabuľky. Text sa zapisuje do riadkov zľava doprava.

15	9	14	13	5	8	6	4	12	10	2	11	16	7	3	1
J	A	-	S	O	M	-	N	I	E	-	O	P	I	L	Y
.	Z	D	R	Ž	I	M	-	V	E	L	M	I	-	M	O
C	.	U	-	B	A	Č	I	K	A	-	P	O	P	I	J
E	M	-	T	O	L	K	O	-	M	O	C	.	A	-	K
T	O	-	J	E	-	T	O	-	T	E	N	/	A		

JA-SOM-NIE-OPILY.ZDRŽIM-VELMI-MOC.U-BAČIKA-POPIJEM-TOLKO-MOC.
A-KTO-JE-TO-TEN/A

[2]V nasledujúcom príklade šifrovania porušíme pravidlá o delení šifrovného textu na časti rôznej dĺžky, a niektoré z depeší ukončíme v strede slova. Toto nám umožní naše šifrované správy hneď v ďalšej časti rozlúštiť.

15	9	14	13	5	8	6	4	12	10	2	11	16	7	3	1
A	/	B	A	Č	I	K	?	D	E	K	A	N	-	V	-
C	H	O	C	H	O	L	O	V	E	.	T	E	N	?	T
O	H	O	-	Z	N	A	M	.	O	D	P	O	V	E	D
A	L	-	O	N	D	R	E	J	-	N	A	R	A	Z	-
V	A	Ž	N	E	.	A	D	U	Š	O	V	/	B		

A/BAČIK?DEKAN-V-CHOCHOLOVE.TEN?TOHO-ZNAM.
ODPOVEDAL-ONDREJ-NARAZ-VAŽNE.ADUŠOV/B

15	9	14	13	5	8	6	4	12	10	2	11	16	7	3	1
B	/	O	-	Z	D	E	L	E	N	I	E	-	U	R	O
B	I	L	O	-	N	A	N	-	V	E	L	K	Y	-	D
O	J	E	M	.	Z	N	A	Š	-	H	O	?	A	K	O
?	K	U	P	I	L	-	S	O	M	-	O	D	-	N	E
H	O	-V	T	E	J	T	O	-	J	E	S	E	N		

B/O-ZDELENIE-UROBILO-NAN-VELKY-DOJEM.ZNAŠ-HO?
AKO?KUPIL-SOM-ODNEHO-TEJTO-JESEN

Z prvých transpozičných tabuliek, text prepíšeme do druhých podľa vyčísleného hesla po stĺpcoch. Tieto stĺpce zapisujeme do riadkov.

Y	O	J	K	-	L	-	O	E	L	M	I	-	N	-	I	O
O	O	Ž	B	O	E	-	M	Č	K	T	I	-	P	A	A	M
I	A	L	-	A	Z	.	M	O	E	E	A	M	T	O	M	P
C	N	I	V	K	-	-	S	R	-	T	J	-	D	U	-	-
J	.	C	E	T	P	I	O	.	/							

JA-SOM-NIE-OPILY.ZDRŽIM-VELMI-MOC.U-BAČIKA-POPIJEM-TOLKO-MOC.
A-KTO-JE-TO-TEN/A

9	2	6	10	7	16	11	13	5	14	3	12	15	4	8	1	17
-	T	D	-	K	.	D	N	O	V	?	E	Z	?	O	M	E
D	Č	H	Z	N	E	K	L	A	R	A	-	N	V	A	B	I
O	N	D	.	/	H	H	L	A	E	E	O	-	Š	A	T	P
A	V	D	V	.	J	U	A	C	-	O	N	B	O	O	-	Ž
A	C	O	A	V	N	E	O	R	/							

A/BAČIK?DEKAN-V-CHOCHOLOVE.TEN?TOHO-ZNAM.
ODPOVEDAL-ONDREJ-NARAZ-VAŽNE.ADUŠOV/B

9	2	6	10	7	16	11	13	5	14	3	12	15	4	8	1	17
O	D	O	E	I	E	H	-	E	R	-	K	N	L	N	A	S
O	Z	-	.	I	E	E	A	N	-	T	U	Y	A	-	N	D
N	Z	L	J	/	I	J	K	O	N	V	-	M	J	E	L	O
O	S	E	-	Š	O	-	-	O	M	P	T	O	L	E	U	-
B	B	O	?	H	-	K	?	D	E							

B/O-ZDELENIE-UROBILO-NAN-VELKY-DOJEM.ZNAŠ-HO?
AKO?KUPIL-SOM-OD-NEHO-TEJTO-JESEN

Z druhých transpozičných tabuliek opäť vypíšeme text podľa vyčísleného hesla po stĺpcoch.

```
IAM-OOAN.MTETNPTDEČOR.JŽLIC-OAKT-AOUYOICJKB-VE--.
-IIIAJOMMSOLKE-/--M-LEZ-POMP-

MBT-TČNVC?AEO?VŠOOAACRDHDDOKN/.VOAAO-DOAA-Z.
VADKHUEE-ONNLLAOVRE-/ZN-B.EHJNEIPŽ

ANLUDZZSB-TVPLAJLENOODO-LEOII/ŠHN-EEOONOBE.
J-?HEJ-KKU-T-AK-?R-NMENYMOEEIO-SDO-
```

Na záver potrebujeme znaky substituovať za čísla. Vytvoríme si teda substitučnú tabuľku pre tretí deň v mesiaci.

	0	1	2	3	4	5	6	7	8	9
0		9	0	A	B	C	Č	D	E	Ě
1	F	G	H	I	J	K	L	M	N	O
2	P	Q	R	Ř	S	Š	T	U	V	W
3	X	Y	Z	Ž	.	?	-	/	1	2
4	3	4	5	6	7	8				

Teraz môžeme znaky zo správy zameniť na číselné kódy podľa tabuľky. Výsledný text sa začína návestím v tvare xxx-yyy-zz, kde xxx predstavuje poradové číslo správy, yyy počet znakov a zz deň šifrovania. Správa sa vždy zapisuje po 5 cifier a ak na konci budú nejaké cifry chýbať, doplníme ich náhodne tak, aby na mieste desiatok boli čísla, ktoré sa tam podľa substitučnej tabuľky nemôžu vyskytnúť (5,6,7,8,9).

```
012-160-03
13031 73619 19031 83417 26082 61820 26070 80619 22341 43316 13053
61903 15263 60319 27311 91305 14150 43628 08363 63436 13131 30314
19171 72419 16150 83637 36361 73616 08323 62019 17203 68063

036-160-03
17042 63626 06182 80535 03081 93528 25191 90303 05220 71207 07191
51837 34281 90303 19360 71903 03363 23428 03071 51227 08083 61918
18161 60319 28220 83637 32183 60434 08121 41808 13203 37391

040-160-03
03181 62707 32322 40436 26282 01603 14160 81819 19071 93616 08191
31337 25121 83608 08191 91819 04083 41436 35120 81436 15151 73626
36031 53635 22361 81708 18311 71908 08131 93624 07193 67287
```

2 Lúštenie TTS

Podrobný popis lúštenia správ zašifrovaných pomocou šifry TTS môžme nájsť v [3] na str. 253-268. V tomto článku sa stručne pokúsim opísať postup, ktorý používali aj nemeckí lúštitelia počas 2. svetovej vojny.

Predstavme si, že sme zachytili tri depeše dĺžky 160 znakov, šifrované v ten istý deň:

```
012-160-03
13031 73619 19031 83417 26082 61820 26070 80619 22341 43316 13053
61903 15263 60319 27311 91305 14150 43628 08363 63436 13131 30314
19171 72419 16150 83637 36361 73616 08323 62019 17203 68063

036-160-03
17042 63626 06182 80535 03081 93528 25191 90303 05220 71207 07191
51837 34281 90303 19360 71903 03363 23428 03071 51227 08083 61918
18161 60319 28220 83637 32183 60434 08121 41808 13203 37391

040-160-03
03181 62707 32322 40436 26282 01603 14160 81819 19071 93616 08191
31337 25121 83608 08191 91819 04083 41436 35120 81436 15151 73626
36031 53635 22361 81708 18311 71908 08131 93624 07193 67287
```

Po obdržaní takýchto textov musíme ako prvú urobiť frekvenčnú analýzu – viď tabuľka 1 na strane 7.

Keď si pozorne prezrieme jednotlivé depeše, zistíme, že posledné dve dvojice číslic majú na mieste desiatok čísla väčšie ako 4. Z toho vyplýva, že tieto dvojice boli doplnené až po zašifrovaní správ, aby tvorili úplnú päticu.

Znalosť frekvenčných tabuliek jazykov (v tomto prípade slovenčiny) nám umožňuje odhadnúť použitú substitúciu. Vieme, že čísla sú v bežnom texte málo používané, je ich desať a nasledujú za sebou. Ďalej vieme, že písmená A, E, O sú veľmi frekventované, a že abeceda v substitučnej tabuľke je zoradená a cyklicky posunutá. Posledným potvrdzujúcim kritériom je medzera, ktorá má najvyšší výskyt spomedzi všetkých znakov. Vieme preto povedať, že substitučná tabuľka vyzerá nasledovne:

	0	1	2	3	4	5	6	7	8	9
0		9	0	A	B	C	Č	D	E	Ě
1	F	G	H	I	J	K	L	M	N	O
2	P	Q	R	Ř	S	Š	T	U	V	W
3	X	Y	Z	Ž	.	?	-	/	1	2
4	3	4	5	6	7	8				

Teraz môžeme vykonať spätnú substitúciu. Je vhodné texty jednotlivých depeší zapísať na štvorčekový papier do riadkov pod seba a jednotlivé stĺpce očíslovať. Umožňuje to depeše lúštiť anagramovou metódou, keďže na všetky tri bola použitá rovnaká transpozícia.

Pri lúštení tohto druhu šifry je najlepšie poznať odosielateľa alebo časť správy. Počas druhej svetovej vojny nemeckí lúštitelia využívali fakt, že depeše odosielané československou vládou mali často sa opakujúci začiatok. V našom príklade môžeme využiť poznatok, že odosielateľ obľubuje klasickú slovenskú literatúru. Mnoho diel v texte obsahuje svoj názov alebo je meno hlavnej postavy zahrnuté v názve diela.

Môže to chvíľu trvať ale podarí sa nám v druhom riadku odhaliť text BAČIK zostavený zo stĺpcov 69, 59, 6, 76 a 48. Toto potvrdzuje našu domnienku a umožňuje pokračovať ďalej. Vieme, že Kukučínov báčik bol z Chochoľova. Preto sa toto

znak	01	02	03	04	05	06	07	08	09	10	11	12	13
výskyt	0	0	17	5	4	2	9	21	0	0	0	5	10
%	0	0	7,08	2,08	1,66	0,83	3,75	8,75	0	0	0	2,08	4,16

znak	14	15	16	17	18	19	20	21	22	23	24	25	26
výskyt	7	8	9	10	14	28	5	0	4	0	3	2	8
%	2,91	3,33	3,75	4,16	5,83	11,66	2,08	0	1,66	0	1,25	0,83	3,33

znak	27	28	29	30	31	32	33	34	35	36	37	38	39
výskyt	3	7	0	0	2	5	1	7	4	30	4	0	0
%	1,25	2,91	0	0	0,85	2,08	0,41	2,91	1,66	12,5	1,66	0	0

znak	40	41	42	43	44	45							
výskyt	0	0	0	0	0	0							
%	0	0	0	0	0	0							

Tabuľka 1: Frekvenčná analýza znakov

slovo pokusíme v niektorom z riadkov poskladať. Objavíme ho opäť v druhom riadku v stĺpcoch 9, 72, 17, 21, 24, 39, 57, 33, 45, 12.

Skontrolovať to môžme veľmi jednoducho. Pokiaľ v jednom z riadkov poskladáme zmysluplné slovo, v ostatných riadkoch sa taktiež musí objaviť zmysluplný text. Ak sa neobjaví, vieme, že musíme vyskúšať inú permutáciu stĺpcov.

Ďalší poznatok vyplývajúci zo samotnej šifry TTS využijeme pri určení druhého a predposledného stĺpca. V obidvoch týchto stĺpcoch musíme mať pod sebou znaky „/“. Výnimočným prípadom sú prvé a posledné depeše v sérii.

Po rozlúštení textu zistíme, že sme mali šťastie, pretože šifrovateľ urobil chybu a rozdelil jednotlivé časti jednej správny na depeše rovnakej dĺžky. Môžeme si preto jednoducho prečítať otvorený text, ktorý predstavuje úryvok z Kukučínovho diela *Keď báčik z Chocholova umrie*:

```
JA-SOM-NIE-OPILY.ZDRŽIM-VELMI-MOC.U-BAČIKA-POPIJEM-TOLKO-MOC.
                              A-KTO-JE-TO-TEN/A

                A/BAČIK?DEKAN-V-CHOCHOLOVE.TEN?TOHO-ZNAM.
                 ODPOVEDAL-ONDREJ-NARAZ- VAŽNE.ADUŠOV/B

            B/O-ZDELENIE-UROBILO-NAN-VELKY-DOJEM.ZNAŠ-HO?
                  AKO?KUPIL-SOM-OD-NEHO- TEJTO-JESEN
```

Pre kontrolu uvádzam postupnosť stĺcov anagramového sledu: 41, 30, 69, 59, 6, 76, 48, 14, 26, 63, 28, 35, 74, 53, 61, 37, 9, 72, 17, 21, 24, 39, 57, 33, 45, 12, 70, 3, 51, 67, 10, 5, 27, 49, 36, 64, 43, 7, 19, 1, 31, 54, 47, 77, 60, 15, 52, 23, 46, 58, 4, 13, 29, 25, 62, 75, 73, 68, 56, 40, 22, 34, 66, 42, 32, 20, 78, 55, 71, 44, 11, 38, 50, 16, 18, 8, 65, 2.

Z tejto postupnosti veľmi jednoducho vytvoríme kryptografický sled tak, že pod čísla anagramového sledu si zapíšeme čísla od 1 po 78 a usporiadame podľa čísel anagramového sledu. Dostaneme nasledovnú postupnosť: 40, 78, 28, 51, 32, 5, 38, 76, 17, 31, 71, 26, 52, 8, 46, 74, 19, 75, 39, 66, 20, 61, 48, 21, 54, 9, 33, 11, 53, 2, 41, 65, 24, 62, 12, 35, 16, 72, 22, 60, 1, 64, 37, 70, 25, 49, 43, 7, 34, 73, 29, 47, 14, 42, 68, 59, 23, 50, 4, 45, 15, 55, 10, 36, 77, 63, 30, 58, 3, 27, 69, 18, 57, 13, 56, 6, 44, 67.

Kryptografický sled nám pomôže odhaliť heslá transpozičných tabuliek. Tento postup je uvedený v [3] str. 265-268.

3 Implementácia

Úlohou mojej bakalárskej práce bolo zostaviť program, ktorý by šifru TTS lúštil automaticky po zadaní niekoľkých textov rovnakej dĺžky.

3.1 Účelová funkcia

Pred samotným testovaním lúštenia si musíme vytvoriť funkciu, ktorá bude vytváraný rozlúštený text hodnotiť. Vstup tejto funkcie bude tvoriť text zložený z niekoľkých správ alebo ich častí a referenčné bi-, tri- a tetra-gramy, ktoré sme získali z dostatočne veľkého textu v danom jazyku. Samotná funkcia vracia bodové hodnotenie vstupného textu podľa toho, ako sa n-gramy v ňom, zhodujú s referenciou.

Ako už bolo spomenuté, referenciu tvoria bi-, tri- a tetra-gramy získané z niekoľkých literárnych diel v slovenčine. Bigramov je teoreticky len 26^2 takže tie

sme pri implementácii v účelovej funkcii použili všetky. Tri- a tetragramov je ale príliš mnoho (teoreticky 26^3 resp. 26^4), a preto sme v účelovej funkcii použili len najčastejších 1000 z nich. Slovom „teoreticky“ v predošlých dvoch vetách sa myslí to, že nie každý možný n-gram sa v použitých vzorkách skutočne vyskytol. Nasledujúca tabuľka zobrazuje skutočné počty bi-, tri- a tetragramov a ich relatívne početnosti v rámci celej vzorky:

	celkovo	početnosť	použité	početnosť
bigramy	607	100%	607	100%
trigramy	8700	100%	1000	75%
tetragramy	78500	100%	1000	63%

3.2 Teória

Zadanie znie jednoducho, poďme sa na to ale pozrieť z matematickej stránky. Janečkov postup vyžaduje presúvanie stĺpcov a hľadanie zmysluplného textu. Počet stĺpcov našich odchytených textov si preto označíme n. Aby sme mohli spoľahlivo určiť zmysluplný text musíme, otestovať všetky možné permutácie týchto stĺpcov. Čiže:

$$n \times (n-1) \times (n-2) \times ... \times 1 = n!$$

Keby sa teda naše n rovnalo 30 musíme otestovať

$$30! = 2,652528598121910586363084 8 \times 10^{32}$$

možností. Tento počet je na praktické testovanie príliš veľký a nasledujúce vyskúšané postupy nie sú schopné ho nijako obmedziť

3.3 Postup č.1: Skladanie podľa bi-gramov

Prvou možnosťou je vybrať jeden štartovací stĺpec, pre ktorý vyberieme zo všetkých ostatných ďalší podľa referencie bi-gramov. Tretí k nim opäť pridáme podľa bi-gramov atď. až kým nepoužijeme všetky stĺpce zo vstupného textu.

V praxi to znamená, že k prvému stĺpcu vyberieme jeden zo zvyšných stĺpcov tak, aby sme dostali, čo najvyššie ohodnotenie. K týmto dvom stĺcom pridáme tretí opäť s cieľom zvýšiť hodnotu účelovej funkcie. Takto postupujeme, kým nepoužijeme všetky stĺpce vstupného šifrovaného textu. Tento postup opakujeme s druhý stĺpcom, tretím stĺpcom atď.
Výpočtová zložitosť tohto algoritmu je rovná:

$$O(n^3)$$

kde `n = počet znakov vstupného textu`.

Postup č. 1 sa ukázal málo účinný a výsledné texty sa vôbec nepodobali zmysluplnému textu.

3.4 Postup č.2: Skladanie podľa bi-gramov, tri-gramov a tetra-gramov

Druhou možnosťou je pre daný štartovací stĺpec vybrať nasledujúci podľa referenčných bi-gramov, tretí podľa tri-gramov a ďalšie podľa tetra-gramov.

Tento spôsob vytvárania permutácií stĺpcov vstupného textu je len obmenou postupu č. 1. Jediný rozdiel nájdeme v hodnotení danej permutácie. Pokiaľ sme už pridali tri stĺpce, naše hodnotenie sa odvíja od referenčných tri-gramov. Pre štyri a viac stĺpcov hodnotíme podľa tetra-gramov. Dôvod pre takéto hodnotenie je prirodzený: tri-gramy obsahujú bi-gramy a tetragramy obsahujú aj bi-gramy aj trigramy.
Výpočtová zložitosť algoritmu postupu č.2 je opäť:

$$O(n^3)$$

kde `n = počet znakov vstupného textu`.

Aj druhý postup bol neúčinný aj keď, výstupný text naozaj začínal vyzerať dobre a medzery so skupinami znakov sa striedali približne, tak ako by to malo byť v obyčajnom texte.

3.5 Postup č.3: Najlepšie prvé tri stĺpce a doplnenie podľa tetra-gramov

Ďalším testovaným postupom skladania stĺpcov bolo zloženie všetkých možností dvoch stĺpcov k danému štartovaciemu a dolpnenie tejto trojice pomocou tetra-gramov.

Tento postup začína vyskladaním všetkých možností dvojice stĺpcov k danému štartovaciemu stĺpcu. Počet týchto možností je ľahko odvoditeľný:

$$p = (n-1) \times (n-2)$$

A algoritmus vyhľadávania najlepšej trojice stĺpcov má výpočtovú zložitosť:

$$O(n^2)$$

pre kde `n = počet znakov vstupného textu`.

Zo všetkých možných dvojíc stĺpcov k nášmu štartovaciemu vyberieme najlepšiu a doskladáme ju podľa tetra-gramov. Algoritmus dohľadania zvyšku permutácie má výpočtovú zložitosť:

$$O(n^2)$$

v tomto prípade `n = počet znakov vstupného textu - 3`, pretože tri stĺpce sme už použili.
Napriek tomu, že v niektorých riadkoch sa objavilo zmysluplné slovo, zvyšok textu bol úplne nezrozumiteľný, preto aj tento postup musím označiť za neúspešný.

3.6 Postup č. 4: Najlepšie prvé štyri stĺpce a ich doplnenie

Posledným testovaným postupom bolo zloženie všetkých možností troch stĺpcov k danému štartovaciemu a doplnenie týchto stĺpcov zvyšnými stĺpcami vstupného textu.

Tento postup je obmednou postupu č. 3. Na začiatku si vyberieme štartovací stĺpec, ku ktorému vyberieme ďalšie tri stĺpce podľa referencie tetra-gramov. Počet všetkých týchto štvoríc je rovný:

$$p = (n-1) \times (n-2) \times (n-3)$$

A algoritmus vyhľadávania najlepšej trojice stĺpcov má výpočtovú zložitosť:

$$O(n^3)$$

kde `n = počet znakov vstupného textu`.

Takto vzniknutý základ permutácie doplníme ostatnými stĺpcami šifrovaného textu tak, aby sa maximalizovala účelová funckia. Algoritmus dohľadania zvyšku permutácie má výpočtovú zložitosť opäť:

$$O(n^2)$$

n sa opäť zmenšilo na `n = počet znakov vstupného textu - 4`, pretože z permutovania vypadli štyri stĺpce, ktoré sme už využili.

3.7 Ďalšie postupy

Existuje samozrejme viac spôsobov ako môžeme problém lúštenia všeobecnej transpozície riešiť. Jedným z nich je aj vyhľadávanie kľúčového slova, ktoré zadá užívateľ.

Zadané slovo hľadáme vo všetkých riadkoch vstupného textu. Odhalenie stĺpcov, v ktorých sa toto slovo nachádza nám zároveň odhalí časti textu v ostatných riadkoch. Čím viac slov užívateľ zadá, tým väčšiu úspešnosť lúštenia môžme predpokladať.

Zložitosť tohto algoritmu závisí od toho, koľko slov zo šifrovaného textu poznáme a aké dlhé depeše odchytíme. Ak by sme odchytili depeše dlhé 40 znakov a poznáme v nich jedno slovo, ktoré má 5 znakov, potrebovali by sme odhaliť už len 35 znakov. To znamená:

$$35! = 1,033\,314\,796\,638\,614\,492\,966\,665\,133\,752\,3 \times 10^{40}$$

Pokiaľ poznáme dve slová, z ktorých jedno má 5 znakov a druhé 6 znakov existuje viacero možností. Môžu sa prekrývať úplne, čo nám odhalí iba o jeden stĺpec viac ako v predchádzajúcom príklade. Nemusia sa však prekrývať vôbec a to nám už odhalí 11 stĺpcov hľadanej permutácie. Zvyšných 29 stĺpcov budeme musieť opäť nájsť. Tzn. počet permutácií na odskúšanie sa rovná:

$$29! = 8\,841\,761\,993\,739\,701\,954\,543\,616\,000\,000$$

Do úvahy musíme samozrejme zobrať aj počet stĺpcov, v ktorých sa vyskytujú rovnaké písmená v tom istom riadku a patria do nášho kľúčového slova. Ak by sme napr. hľadali slovo LONDÝN a písmeno L by sa v šifrovanom texte vyskytovalo dvakrát a písmeno N by sa v šifrovanom texte vyskytovalo tri krát máme

$$2 \times 1 \times 3 \times 1 \times 1 \times 2 = 12$$

možností, ktorými toto slovo poskladať. Správnosť nášho pokusu nám dokáže len overenie textu v ostatných riadkoch.

Tento postup som netestoval ale dá sa predpokladať oveľa väčšia úspešnosť ako v predchádzajúcich prípadoch.

3.8 Obmedzenia

Pokiaľ túto šifru lúšti človek, nie je to pre neho ťažká úloha. Ľudský mozog pracuje ako obrovský slovník a superparalelný počítač, ktorý okamžite spozná asociácie medzi slovami a písmenami v nich. Zároveň má skúsenosti so stavbou jazyka textu, ktorý lúšti a pozná nepreberné množstvo slovných tvarov a možností ako sú tieto slová spojené do viet. Takýto postup však nie je ľahké implementovať. Musíme si preto stanoviť rôzne obmedzujúce podmienky, aby sme boli schopní šifru TTS s bežne dostupným výpočtovým výkonom lúštiť.

Príkladom takéhoto obmedzenia by mohol byť dátum na začiatku každej depeše v tvare `DD-MM-YYYY`. Takýto blok textu je ľahko identifikovateľný a pokiaľ by sa vyskytoval len v každej tretej depeši správy, pomohol by odhaliť aj prvých 10 znakov ostatných depeší, ktoré sme odchytili. Ak by sa vyskytoval aj na konci depeše bolo by to už 20 stĺpcov.

V strede medzi takto pevne určeným blokmi textu nám ale zostane ešte niekoľko stĺpcov. Tieto stĺpce predstavujú permutáciu, ktorú musíme odhaliť, pretože neposkytujú žiadne pevné vodítko. Pre realizáciu na bežnom počítači by počet týchto neodhalených stĺpcov nemal byť väčší než 13, pretože:

$$13! = 6\,227\,020\,800$$

Toto číslo môžme považovať za rozumné a preto budeme testovať 6 227 020 800 možností, z ktorých v jednej sa určite vyskytne zmysluplný text vo všetkých riadkoch.

Týmto spôsobom môžeme lúštiť depeše dlhé 33 znakov. Ak by sme chceli lúštiť dlhšie texty, museli by sme stanoviť ďalšie obmedzujúce podmienky.

Literatúra

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

[2] Jiří Janeček: *Gentlemani nečtou cizí dopisy*, Books Bonus A, 1998

[3] Jiří Janeček: *Válka šifer- výhry a prohry československé vojenské rozvědky*, Votobia, 2001

[4] Jozef Kollár: *Československé šifry z obdobia 2. svetovej vojny Diel 1., Šifra TTS*, Cryptoworld 1, 2011

B. Nový (souhrnný) pohled na otázky bezpečnosti eliptické kryptografie (překlad a výběr z informací provedl Jaroslav Pinkava, jaroslav.pinkava@gmail.com)

Zajímavé a aktuální vystoupení (Daniel J. Bernstein and Tanja Lange: " *SafeCurves: Choosing Safe Curves for Elliptic-Curve Cryptography"* [1]) dvou matematických odborníků (publikují články zaměřené na matematické problémy kryptologie) na konferenci Shmoocon (probíhala ve dnech 17 až 19. ledna 2014 ve Washingtonu - [4]) se týká situace v eliptické kryptografii.

Existuje celá řada různých norem, podle nichž si lze zvolit křivky určené pro použití v eliptické kryptografii (např. ANSI X9.62, IEEE P1363, SEC 2, NIST FIPS 186-2, ANSI X9.63, Brainpool, NSA Suite B a ANSSI FRP256V1 - viz odkazy uvedené v [2]). Každá z těchto norem se pokouší ujistit, že řešení problému eliptického diskrétního logaritmu (ECDLP) je matematicky obtížnou úlohou. ECDLP znamená problém nalézt uživatelův soukromý (utajovaný) klíč při daném jeho veřejném klíči. Autoři však podotýkají, že bohužel existuje rozdíl mezi obtížností ECDLP a bezpečností ECC (eliptic curve cryptography). Bezpečné implementace eliptických křivek z norem jsou teoreticky možné, ale velice obtížné. Chybovat je možné ve čtyřech směrech:

- vaše implementace vede k nesprávným výsledkům pro některé vzácné body na křivce;
- vaše implementace umožňuje únik utajovaných dat pokud vstupem není bod na křivce;
- vaše implementace umožňuje únik utajovaných dat při sledování časů (timing) v jednotlivých větvích algoritmu;
- vaše implementace umožňuje únik utajovaných dat při sledování časů cache.

Autoři uvádí následující rozdíly mezi ECDLP a ECC v reálném světě, ty které jsou útočníky využívány:

- ECDLP je neinteraktivní. Reálná ECC pojednává vstupy kontrolované útočníky;
- ECDLP dává pouze nP. Reálná ECC má jako výstup i časy (a někdy i další informace z postranních kanálů);
- ECDLP vždy spočte nP správně. Reálná ECC někdy chybuje.

Tyto problémy byly již útočníky reálně využity a vedly proto autory k návrhu tzv. **SafeCurves**. Kritéria pro jejich výběr zahrnují i bezpečnost ve vztahu k ECC, tedy nikoliv pouze ve vztahu k bezpečnosti ECDLP. Ke křivkám, které jsou zmiňovány v citovaných normách, byla přidávána často další omezení - některá z nich však bohužel vedou k horší bezpečnosti.

Na stránce [2] se lze seznámit s ohodnocením řady těchto křivek z norem (popřípadě křivek uvedených v publikacích některých autorů).

Touto problematikou se autoři dále zabývají na dalších podstránkách v druhém odkazu. Hlouběji se zabývají nejprve problémem bezpečnosti ECDLP - například na stránce [3] zavádí pojem "*Rigidity*", což je vlastnost související s generováním křivky, kde počet generovaných křivek je omezen.

Co se pak týká bezpečnosti ECC, autoři rozdělují poznatky z literatury a své poznatky celkem do čtyř odstavců (každý má oddělenou stránku): *Ladders, Twists. Completeness, Indistinguishability.* Podle kritérií, která jsou uvedena v těchto odstavcích, jsou pak v tabulkách uvedena hodnocení jednotlivých křivek. Na stránce *References* pak autoři uvádí rozsáhlou bibliografii k problémům eliptické kryptografie. Na závěrečné stránce *Verification* je pak uveden skript umožňující ověřit kritéria pro SafeCurves - zda jsou naplněna pro danou křivku (její ECC implementaci).

Literatura:

[1] https://archive.org/details/ShmooCon2014_SafeCurves

[2] http://safecurves.cr.yp.to/ (Daniel J. Bernstein and Tanja Lange: SafeCurves: choosing safe curves for elliptic-curve cryptography)

[3] http://safecurves.cr.yp.to/rigid.html

[4] http://www.shmoocon.org/

C. Vyhláška o kybernetické bezpečnosti – výzva k připomínkám

pravni@nbu.cz

a) NBÚ na svém webu zveřejnil 9.1.2014 následující informaci:

Vláda 2. ledna 2014 schválila Návrh zákona o kybernetické bezpečnosti

Vláda České republiky ve čtvrtek 2. ledna 2014 schválila Návrh zákona o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti).

http://www.vlada.cz/cz/media-centrum/tiskove-zpravy/vysledky-jednani-vlady--2--ledna-2014-114800/

Tento návrh zákona připravil a předložil Národní bezpečnostní úřad. Návrh zákona bude předložen k dalšímu legislativnímu projednávání v Parlamentu České republiky. Aktuální návrh zákona včetně důvodové zprávy můžete stáhnout zde.

http://www.govcert.cz/cs/informacni-servis/aktuality/vlada-2-ledna-2014-schvalila-navrh-zakona-o-kyberneticke-bezpecnosti/

b) Koncem února pak NBÚ zveřejnil následující sdělení obsahující výzvu směřovanou k odoborné veřejnosti s žádostí o zaslání připomínek k návrhu textu vyhlášky:

NBÚ vypracoval návrh vyhlášky o kybernetické bezpečnosti

21.02.2014

Národní bezpečnostní úřad vypracoval k návrhu zákona o kybernetické bezpečnosti, který byl předložen k dalšímu legislativnímu procesu do Parlamentu České republiky, návrh prováděcího předpisu, kterým je vyhláška o bezpečnostních opatřeních, kybernetických bezpečnostních incidentech, reaktivních opatřeních a o stanovení náležitostí podání v oblasti kybernetické bezpečnosti (vyhláška o kybernetické bezpečnosti). Návrh vyhlášky o kybernetické bezpečnosti zejména naplňuje a rozvádí první pilíř zákona o kybernetické bezpečnosti - bezpečnostní opatření, neboli požadavky na standardizaci kritické informační infrastruktury a významných informačních systémů.

Případné připomínky odborné veřejnosti k návrhu vyhlášky lze uplatnit do 17. března 2014 na adrese pravni@nbu.cz .

Návrh vyhlášky o kybernetické bezpečnosti (Návrh pro vnější připomínkové řízení) můžete stáhnout zde.

http://www.nbu.cz/download/nodeid-912/

D. Několik poznámek ke kryptografickým požadavkům uvedeným ve vyhlášce o kybernetické bezpečnosti

Pavel Vondruška, pavel.vondruska@crypto-world.info

V návaznosti na výzvu NBÚ k odborné veřejnosti k zaslání připomínek k návrhu vyhlášky o kybernetické bezpečnosti, jsem připravil několik svých poznámek k části, které se zabývá požadavky na kryptografické prostředky (§ 25) a k minimálním požadavkům na kryptografické algoritmy, které jsou uvedeny v příloze č. 3 k této vyhlášce.

Nejprve se podívejme na znění příslušného paragrafu a odpovídající odkazované vyhlášky:

§25
Kryptografické prostředky

(1) Orgán a osoba uvedená v § 3 písm. c) až e) zákona splní povinnost podle § 4 odst. 2 zákona tím, že pro používání kryptografické ochrany, stanoví

a) úroveň ochrany s ohledem na typ a sílu kryptografického algoritmu a

b) pravidla kryptografické ochrany informací při přenosu po komunikačních sítích nebo při uložení na mobilní zařízení nebo vyměnitelná média.

(2) Orgán a osoba uvedená v § 3 písm. c) až e) zákona splní povinnost podle § 4 odst. 2 zákona tím, že používá kryptografické prostředky, které zajistí ochranu důvěrnosti a integrity předávaných nebo ukládaných dat a prokázání odpovědnosti za provedené činnosti.

(3) Orgán a osoba uvedená v § 3 písm. c) a d) zákona splní povinnost podle § 4 odst. 2 zákona tím, že

a) stanoví pro používání kryptografických prostředků systém správy klíčů, který zajistí generování, distribuci, ukládání, archivaci, změny, ničení, kontrolu a audit klíčů a

b) používá odolné kryptografické algoritmy a kryptografické klíče; minimální požadavky na kryptografické algoritmy jsou uvedeny v příloze č. 3 k této vyhlášce.

Příloha č. 3 k vyhlášce č. …/2014

Minimální požadavky na kryptografické algoritmy

a) Symetrické algoritmy

1) Blokové a proudové šifry pro ochranu důvěrnosti a integrity

a. Advanced Encryption Standard (AES) s využitím délky klíčů 128, 192 a 256 bitů Triple Data Encryption Standard (3DES) s využitím délky klíčů 168 bitů, omezené použití jen se zatížením klíče menším než 10 GB, postupně přecházet na AES.

b. Blowfish s využitím minimální délky klíčů 128 bitů, omezené použití jen se zatížením klíče menším než 10 GB.

c. Kasumi s využitím délky klíčů 128 bitů, omezené použití jen se zatížením klíče menším než 10 GB.

d. RC 4 s využitím minimální délky klíčů 128 bitů.

e. SNOW 2.0, SNOW 3G s využitím délky klíčů 128, 256 bitů.

f. Twofish s využitím délky klíčů 128 až 256 bitů.

g. Serpent s využitím délky klíčů 128, 192, 256 bitů.

2) Módy pro ochranu integrity

a. HMAC

b. CBC-MAC-X9.19, omezené použití jen se zatížením menším než 10^9 MAC

c. CBC-MAC-EMAC

d. CMAC

b) Asymetrické algoritmy

1) Pro technologii digitálního podpisu

a. Digital Signature Algorithm (DSA) s využitím délky klíčů 2048 bitů a více, délky parametru cyklické podgrupy 224 bitů a více.

b. Elliptic Curve Digital Signature Algorithm (EC-DSA) s využitím délky klíčů 224 bitů a více.

c. Rivest-Shamir-Adleman Probablistic Signature Scheme (RSA-PSS) s využitím délky klíčů 2048 bitů a více.

2) Pro procesy dohod na klíči a šifrování klíčů

a. Diffie-Hellman (DH) s využitím délky klíčů 2048 bitů a více, délky parametru cyklické podgrupy 224 bitů a více.

b. Elliptic Curve Diffie-Hellman (ECDH) s využitím délky klíčů 224 bitů a více.

c. Elliptic Curve Integrated Encryption System - Key Encapculation Mechanism (ECIES-KEM) s využitím délky klíčů 256 bitů a více.

d. Provably Secure Elliptic Curve - Key Encapculation Mechanism (PSEC-KEM) s využitím délky klíčů 256 bitů více.

e. Asymetric Ciphers and Key Encapculation Mechanism (ACE-KEM) s využitím délky klíčů 256 bitů a více.

f. Rivest Shamir Adleman - Optimal Asymetric Encryption Pedding (RSA-OAEP) s využitím délky klíčů 2048 a více.

g. Rivest Shamir Adleman - Key Encapculation Mechanism (RSA-KEM) s využitím délky klíčů 2048 a více.

c) Algoritmy hash funkcí

1) Secure Hashing Algorithm 2,
2) SHA-224,
3) SHA-256,
4) SHA-384,
5) SHA-512/224
6) SHA-512/256),
7) RIPEMD-160,
8) Whirlpool a
9) SHA-1 nepoužívat v podepisovacích algoritmech.

Několik drobných poznámek k výše uvedeným požadavkům

Lze diskutovat, zda „Orgány a osoby uvedené v § 3 písm. c) až e)“ jsou dostatečně připraveni naplnit požadavky uvedené v §25. V současné praxi nejsou požadavky v bezpečnostní dokumentaci (politiky, bezpečnostní směrnice, manuály administrátorů) specifikovány takto detailně. Zejména odst. 3 obsahuje požadavky, které přesahují běžně používaná specifika v bezpečnostní dokumentaci. Zde uvedené požadavky na klíčové hospodářství jsou sice z hlediska kryptografické bezpečnosti správné, ale dovoluji si tvrdit, že mimo oblast ochrany utajovaných informací pomocí schválených kryptografických prostředků, se takto detailně pro jiné účely klíčové hospodářství nezpracovává. Z praxe vím, že zpravidla manažeři informační bezpečnosti ani nemají k dispozici u dodávaných běžných produktů, které se ke kryptografické ochraně používají, tyto informace k dispozici. Např. otázka generování klíčů (přesněji otázka použitých náhodných generátorů) se nijak samostatně neřeší a prostě se bere na vědomí, že dodaný produkt „klíče generuje“. Také otázka ničení klíčů bývá běžně opomíjena.

Požadavky v celém §25 jsou podle mne rozumné a určitě správné, pouze k jejich zajištění bude nutno vykonat relativně dost administrativní práce v citovaných „orgánech“ a bude nutné udělat „inventuru“ toho, co organizace skutečně mají k dispozici a trochu více se zajímat o kryptografické vlastnosti těchto produktů a aplikací. Bohužel se obávám, že ne

všechny informace mají tyto orgány k dispozici, protože dodavatelé komerčních kryptografických produktů často tyto informace dodávají neúplné a některé detaily (např. vlastnosti generátorů náhodných čísle) ani nemají sami k dispozici nebo jsou dokonce obchodním tajemstvím.

Pokud jde o „Minimální požadavky na kryptografické algoritmy“ uvedené v odkazované příloze č. 3 vyhlášky, pak je situace obdobná. Zde uvedené požadavky jsou z hlediska bezpečnosti voleny rozumně, ale ten, kdo má ověřit, že již v organizaci používané prostředky toto využívají, může mít opět problém s tím to „dokázat“, protože doprovodná dokumentace dodávaných komerčních aplikací a produktů nemusí dát jasnou, dostatečnou a jendoznačnou odpověď.

Jako příklad se uveďme ve vyhlášce uvedený požadavek na použití *Triple Data Encryption Standard* (3DES) s využitím délky klíčů 168 bitů. Z délky klíče 168 bitů plyne, že se tedy vyžaduje algoritmus 3DES s použitím 3 různých klíčů délky 56 bitů. Nepovoluje se tedy v mnoha aplikacích stále běžně využívané použití „dvou klíčů“ (K1, K2, K1) o celkové délce 112 bitů. Z bezpečnostního hlediska je to v pořádku a je to v souladu s obecnými bezpečnostně standardizačními trendy. Tento požadavek byl vysloven ve standardu NIST Special Publication 800-131 - Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths, Januar 2011 (přesněji již v draftu tohoto dokumentu z léta 2012).

Table 1: Encryption Transitions

Algorithm	Use
Two-key Triple DES Encryption	Acceptable through 2010 Restricted use from 2011 through 2015 Disallowed after 2015
Two-key Triple DES Decryption	Acceptable through 2010 Legacy-use after 2010
Three-key Triple DES Encryption and Decryption	Acceptable

Zde se konkrétně vyžaduje, aby byl „ Two-key Triple DES Encryption“ používán bez omezení jen do konce roku 2010, jeho použití od roku 2011 do roku 2015 by mělo být již omezené na situace, kde se nepředpokládá dlouhodobé uchování šifrového obsahu, od roku 2015 by měl být 3DES používán již pouze se „třemi“ klíči.

Ovšem „Studie ENISA: Algorithms, Key Sizes and Parameters Report. 2013 recommendations, 29.10.2013“ se tak striktně na tuto problematiku nedívá. Píše se zde, že použití „dvouklíčového DESu“ není problém v případě, že se generuje klíč na každou zprávu. Důvodem je, že byl popsán útok (v praxi ovšem těžko proveditelný útok) založený na znalosti velkého množství dvojic otevřený/šifrový text. V takovém případě je bezpečnost pro dvouklíčový DES nikoliv 2^{112} ale 2^{120-t} , kde 2^t je počet známých dvojic plaintext/ciphertext.

Není vyloučeno, že právě toto myslí předkladatelé vyhlášky onou větou: *„omezené použití jen se zatížením klíče menším než 10 GB“*. Pokud ano, pak by mělo být jasnější, že pokud je použit „dvouklíčový 3DES“, pak jen se zatížením klíče menším 10 GB.

Navíc opět mohu z praxe uvést, že běžné komerční produkty zpravidla neuvádí, zda je 3DES implementován ve verzi dvou nebo tří klíčové. Uživatel to (zpravidla) nemůže sám jednoduše zjistit nebo ověřit. Z toho plyne, že pro odpovědné osoby bude obtížné tento požadavek tam, kde se 3DES používá ověřit a prokázat.

Pokud zůstaneme stále u odstavce a) přílohy – tj. u požadavků na symetrické blokové algoritmy (šifry), pak se domnívám, že měla být doplněn odstavec o „doporučení“ / požadavky na použití konkrétních blokových modů. Minimálně by bylo vhodné z hlediska bezpečnosti zakázat ECB mód (Electronic Code Book). Ovšem ani použití CBC módu (tedy zcela jednoznačně nejrozšířenějšího módu) není v doporučení výše jmenované studie „Studie ENISA: Algorithms, Key Sizes and Parameters Report. 2013 recommendations“ již považováno pro všechny účely za vhodné.

enisa

Algorithms, Key Size and Parameters Report – 2013

Scheme	IND-CPA	IND-CVA	IND-CCA	
Block Cipher Modes of Operation				
OFB	✓	(✓)	✗	No padding
CFB	✓	(✓)	✗	No padding
CTR	✓	(✓)	✗	No padding
CBC	✓	✗	✗	
ECB	✗	✗	✗	See text
XTS	-	-	✗	See text
EME	-	-	✗	See text

Současně si uvědomuji, že opět osoba uvedená v §25, která bude odpovědná za výběr doporučených algoritmů bude i zde mít nelehkou úlohu, protože opět často u komerčních

produktů není uvedeno, jaké blokové módy jsou reálně použity a není možné je parametricky nastavit.

Pokud jde o výběr asymetrických algoritmů, pak není (opět na rozdíl od studie ENISA) uveden algoritmus Camellia (např. viz Camellia Encryption for Kerberos 5. RFC 6803 nebo OpenSSL a zejména TLS). Znamená to, že uvedené protokoly nebude možné považovat podle této přílohy za „použitelné“?

Závěr / doporuční k symetrickým algoritmům:

- Povolit dvouklíčový 3DES alespoň tam, kde se generuje klíč na každé použití znovu
- Doplnit požadavky na použití blokových módů
- Doplnit algoritmus Camellia

Přeskočme nyní odstavec b) přílohy a podívejme se na odstavec c) Algoritmy hash funkcí.

V souladu se současnými trendy jsou zde uvedeny hashovací funkce třídy SHA-2 (dokonce taxativně jsou vyjmenovány všechny varianty) a ty jsou doplněny o RIPEMD-160, Whirlpool a dokonce ještě o SHA-1 (s poznámkou - mimo použití v podpisových algoritmech).

Ochota použít SHA-1 mimo podpisové algoritmy mne trochu překvapuje. Z jedné strany rozumím tomu, že stále existuje řada aplikací / protokolů, která SHA-1 využívá a její plošný zákaz by byl pro uživatele velkým problémem. Na druhé straně je potřeba si říci, že nalezení kolizí SHA-1 bude mít negativní bezpečnostní dopad i na jiné situace než je použití k podpisům. To se ostatně prokázalo při nalezení kolizí MD-5, kdy krátce po zveřejnění již prvé kolize (v té době jediné) se objevily články, jak lze využít kolizi k různým útokům a to i mimo podpisové (přesněji non-digiatal) aplikace.

U SHA-1 sice ještě kolize objeveny nebyly, ale je to jen otázka času (osobně si dovolím předpovědět, že možná krátkého a věřím, že by mohly být publikovány do konce tohoto roku nejpozději do konce příštího roku.

Pro připomenutí, jaký je aktuální stav hledání kolizí u SHA-1 si uvedeme pár „citací“:

SHA-1 byla kryptograficky prolomena v roce 2005.
prof. Xiaoyun Wang publikovala útoka na algoritmus SHA-1 se složitostí 2^69 .

SHA-1 byla teoreticky prolomena v 9/2005
Akashi Satoh: Hardware Architecture and Cost Estimates for Breaking SHA-1, ISC 2005, Singapore, September 20-23, 2005, LNCS 3650, pp. 259-273, 2005

A.Satoh publikuje v uvedené práci návrh hardware, který by nalezl kolize podle návrhu Wangové se složitostí 2^69. Navrhuje se architektura LSI na bázi 0.13-µm CMOS. Na základě toho byla vypočítána rychlost, velikost a spotřeba HW.
Za 10 milionů dolarů lze sestavit zákaznický hardwarový systém, který by sestával z 303 PC, každý s 16 deskami (na každé je 32 jader SHA-1), pracujícími paralelně. Útok by trval 127 dní.
A dále se útoky samozřejmě již jen budou vylepšovat.
Na rump session Crypto 2005 (17.8.2005), oznámila paní prof. Xiaoyun Wang urychlení svého vlastního útoku na SHA-1 z původní složitosti 2^69 na 2^63.
V roce 2013 pak publikoval Marc Stevens útok (ovšem pouze na hledání „lokálních kolizí“) ze složitostí již jen 2^57.
Marc Stevens. New collision attacks on SHA-1 based on optimal joint local-collision analysis. In Johansson and Nguyen, pages 245-261

V již změněném standardu NIST Special Publication 800-131 - Recommendation for the Transitioning of Cryptographic Algorithms and Key Lengths, Januar 2011, je k SHA-1 a jeho použití toto doporučení:

Table 9: Hash Function Transitions

Hash Function	Use	
SHA-1	Digital signature generation	Acceptable through 2010 Deprecated from 2011 through 2013 Disallowed after 2013
	Digital signature verification	Acceptable through 2010 Legacy-use after 2010
	Non-digital signature generation applications	Acceptable

Non-digital signature generation aplications jsou ve standardu definovány takto: Applications include HMAC, Key Derivation Functions (KDFs), Random Number Generation (RNGs and RBGs), and hash-only applications (e.g., hashing passwords and using SHA-1 to compute a checksum, such as the approved integrity technique specified in Section 4.6.1 of [FIPS 140-2]).

Lze tedy říci, že požadavek využití SHA-1 tak, jak je v příloze k vyhlášce uveden, je z hlediska bezpečnosti v souladu se standardy NIST i EU (ENISA). Přesto bych osobně považoval za vhodné stanovit např. u nově zaváděných systémů, aby tam, kde to je možné již SHA-1 nebyla využívána.

Co mne však překvapilo ještě více je úplná absence nového kryptografického standardu SHA-3, který vzešel ze soutěže pořádané NIST a kterým se stal 2.10.1012 algoritmus Keccak. K dispozici k tomuto algoritmu je bohatý materiál a již je doplňován do aplikačních knihoven. Pokud jde o standardy de jure, pak však existuje zatím pouze draft FIPS SHA-3. Standard FIPS SHA-3 má být publikován koncem II.Q.2014. Tento algoritmus by určitě mezi bezpečnými hashovacími funkcemi neměl chybět. V době platnosti vyhlášky bude implementován v řadě aplikacích a bylo by vhodné, aby mohl být v souladu s vyhláškou použit.

Závěr / doporuční k algoritmům hashovacích funckí:

- Zařadit hashovací funkci SHA-3
- Upřesnit použití hashovací funkce SHA-1 tj. přesněji definovat, co se chápe pod pojmem podpisové algoritmy.

Vyhláška samozřejmě není odborná studie a nemůže proto v prostoru, který jednotlivým aspektům bezpečnosti věnuje se zabývat všemi aspekty. Věřím, že některé výše uvedené problémy / otázky se vyjasní při aplikaci vyhlášky do praxe. Obecně je (podle mne) vyhláška napsána jasně, srozumitelně a je z odborného hlediska na velice dobré úrovni. Její aplikace subjekty, kterých se týká, však bude vyžadovat práci, čas a shromáždění určitého know-how o aplikacích, které byly dosud využívány bez toho, že by byly tak odborně při nasazení posuzovány a jejich uživatelé nemusí mít (a pravděpodobně v řadě případů nemají) k dispozici všechny potřebné informace, které by jim pomohly zodpovědět otázku, zda jsou v souladu s požadavky vyhlášky, speciálně Přílohy 3 k této vyhlášce.

E. O čem jsme psali v předchozích 149 číslech…

Kompletní obsah všech **149** dosud vyšlých čísel od roku 1999 je dostupný zde:
http://crypto-world.info/index2.php?vyber=obsah
http://crypto-world.info/obsah/obsah_roky.pdf

Přehled obsahu posledních vydaných čísel

Crypto-World 11-12/2013

Příloha k článku D: http://web.telecom.cz/depotpv/ASD12/priloha_k_D.zip

Crypto-World 9-10/2013

Příloha: ukázka z knihy Aplikovaná kryptografie
http://crypto-world.info/casop15/Burda_akryptografie.pdf

Crypto-World 7-8/2013

F. 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
	Jozef Krajčovič
	Jozef Martin Kollar
	Vlastimil Klíma
Přehled autorů:	http://crypto-world.info/obsah/autori.pdf
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/
Jozef Martin Kollar	jmkollar@math.sk ,	
Jozef Krajčovič	kryptosvet@gmail.com ,	http://katkryptolog.blogspot.sk
Jaroslav Pinkava	jaroslav.pinkava@gmail.com ,	http://crypto-world.info/pinkava/
Pavel Vondruška	pavel.vondruska@crypto-world.info	http://crypto-world.info/vondruska/index.php
Pavel Vondruška, jr.	pavel@crypto-world.info ,	http://www.pavelvondruska.cz/