Slovenská technická univerzita v Bratislave
Fakulta elektrotechniky a informatiky
Katedra informatiky a výpočtovej techniky

Implementácia blokového šifrátora pomocou PLD

Tímový projekt

team č. 12:

Bc. Martin Prokša
Bc. Viliam Otepka
Bc. Ivan Varga
Bc. Martin Zeman

vedúci projektu: doc. Ing. Ladislav Hudec, CS
šk. rok: 2002/2003

I Dokumentácia k projektu

Obsah

I.1 Úvod

Tento projekt sa zaoberá problematikou šifrovania, prezentácie šifrovacieho algoritmu ako aj samotný návrh hardvérovej implementácie šifry. V tejto časti dokumentu sa budeme venovať iba problematike šifrovania a jej prezentácie.

I.1.1 Rozsah a ciele

Vzhľadom na to, že daná problematika je obšírna je práca na tímovom projekte je rozdelená do dvoch častí. Úlohou zimného semestra je vytvoriť grafickú prezentáciu šifrovania pomocou šifry IDEA [4] a vysvetlenie tejto šifry. Výstupy z tejto časti budú uverejnené na tímovej web stránke. Cieľom vytvárania prezentácie je oboznámiť sa s princípmi šifry IDEA, ako aj priblížiť túto šifru ostatným záujemcom ktorý nemajú žiadne vedomosti z oblasti šifrovania.

V letnom semestri bude naša práca pozostávať z navrhnutia, implementácia a simulácie návrhu šifry IDEA do PLD programovateľného obvodu.

Cieľom tejto časti dokumentu je analyzovať, špecifikovať a navrhnúť interaktívnu prezentáciu šifry IDEA. Budeme sa venovať analýze a návrhu implementácie z pohľadu hardvérovo nezávislých funkcií obvodu, ako napríklad zhodnotenie rôznych prístupov k implementácií matematicko-logických operácií.

I.1.2 Prehľad dokumentu

Tento dokument sa stane súčasťou dokumentácie k tímovému projektu. Dokumenty, ktoré budú tvoriť ucelený celok budú označené rímskymi číslicami.

V tomto dokumente sa nachádzajú špecifikácie požiadaviek na

interaktívnu prezentáciu šifry IDEA. Dokument analyzuje problém šifrovania pomocou šifry IDEA a návrh riešenia interaktívnej prezentácie šifry IDEA.

I.2 Analýza problému

I.2.1 Princípy šifry IDEA

IDEA (International Data Encryption Algorithm) je názov patentovaného a univerzálne aplikovateľného blokového šifrovacieho algoritmu. Používa rovnaký kľúč (tajný kľúč) na šifrovanie aj dešifrovanie. S kľúčom veľkosti 128 bitov, IDEA je ďaleko bezpečnejšia ako široko známy DES (Data Encryption Standard). Mnohí považujú tento algoritmus za najbezpečnejší blokový šifrovací algoritmus týchto dní. Algoritmus bol vytvorený v rámci projektu v spolupráci so Švajčiarskym federálnym technologickým inštitútom a spoločnosťou Ascom [4]. Hlavným cieľom tohoto projektu bolo vytvoriť silný šifrovací algoritmus, ktorý by mal nahradiť DES algoritmus.

Výhody algoritmu:

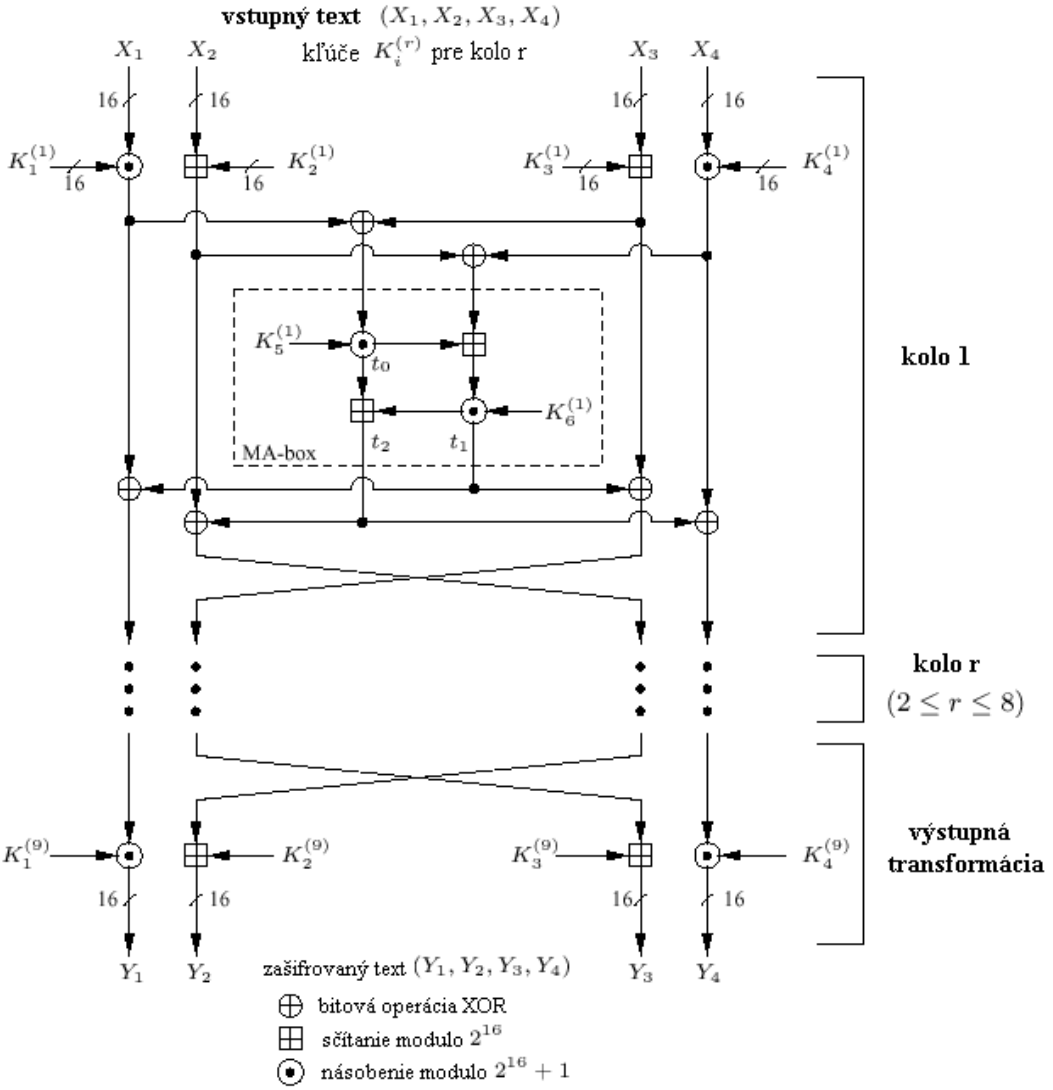
- poskytuje vysoký stupeň ochrany
- je úplne špecifikovaný a jednoducho pochopiteľný
- prístupný pre všetkých
- použiteľný v širokom rozsahu aplikácií
- môže byť ekonomicky implementovateľný do elektronických komponentov (VLSI čip)
- môže byť využitý efektívne
- je silný, malý a rýchly

I.2.2 IDEA šifrovanie

IDEA šifruje 64 bitový vstupný text na 64 bitový zašifrovaný blok, použitím 128 bitového kľúča. Proces šifrovania pozostáva z ôsmich výpočtovo identických kôl a výstupnej transformácie [1]. Výstup z nejakého kola je

vstupom pre nasledujúce kolo. Výstup z ôsmeho kola je vstupom výstupnej transformácie, a jej výstupom je už požadovaný zašifrovaný blok.

Na obrázku č. I.1 sú zobrazené bloky šifry IDEA. Na začiatku je rozkreslené prvé kolo šifrovania. Vstupy X1,X2,X3,X4 sú nezašifrovaný vstupný text rozdelený do štyroch 16 bitových slov. Na konci vystupuje zašifrovaný blok rozdelený do štyroch 16 bitových slov Y1,Y2,Y3,Y4 .



obr č.I.1 Bloky šifry IDEA

Ako je zrejmé z obrázku, do každého kola vstupuje sada šiestich kľúčov veľkosti 16 bitov. Pre osem kôl teda potrebujeme 48 kľúčov. Výstupná transformácia potrebuje naviac štyri kľúče. Spolu je teda potrebné vygenerovať 52 kľúčov.

Tieto šifrovacie kľúče sa generujú zo vstupného 128 bitového kľúča nasledovným spôsobom:

1. 128 bitový vstupný kľúč je rozdelený na osem 16 bitových kľúčov – prvých osem kľúčov
2. 128 bitový kľúč je cyklicky bitovo posunutý vľavo o 25 bitov a znovu rozdelený na osem ďalších 16 bitových kľúčov
3. druhý krok sa opakuje dovtedy, pokiaľ nie je vygenerovaných všetkých 52 kľúčov

Ako je zrejmé z obrázka, algoritmus využíva tri základné operácie:

- bitová operácia XOR – dva 16 bit vstupy, jeden 16 bit výstup
- sčítanie modulo 2^{16} - dva 16 bit vstupy, jeden 16 bit výstup
- násobenie modulo $2^{16} + 1$ (nula je interpretovaná ako 2^{16}) - dva 16 bit vstupy, jeden 16 bit výstup – algoritmus je možné nájsť v [2]

I.2.3 IDEA dešifrovanie

Dešifrovanie využíva rovnaký algoritmus ako šifrovanie, pričom vstup je zašifrovaný 64 bitový blok a výstup je dešifrovaný text. Rozdiel je v tom, že sa použije 52 dešifrovacích kľúčov, ktoré sú odvodené od 52 šifrovacích kľúčov. Transformácia šifrovacích kľúčov na dešifrovacie určuje nasledovná tabuľka:

kolo r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$
$r = 1$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_2$	$-K^{(10-r)}_3$	$(K^{(10-r)}_4)^{-1}$	$K^{(9-r)}_5$	$K^{(9-r)}_6$
$2 \leq r \leq 8$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_3$	$-K^{(10-r)}_2$	$(K^{(10-r)}_4)^{-1}$	$K^{(9-r)}_5$	$K^{(9-r)}_6$
$r = 9$	$(K^{(10-r)}_1)^{-1}$	$-K^{(10-r)}_2$	$-K^{(10-r)}_3$	$(K^{(10-r)}_4)^{-1}$	—	—

obr č.I.2 Transformácia šifrovacích kľúčov na dešifrovacie

Popis indexov k obrázku č.I.2:

r – číslo kola

K – šifrovacie kľúče

K^I – dešifrovacie kľúče

$-K_i$ – inverzné sčítanie (modulo 2^{16}) kľúča K_i

K_i^{-1} – inverzné násobenie (modulo $2^{16} + 1$) kľúča K_i

Inverzné násobenie sa dá odvodiť z rozšíreného Euklidovho algoritmu (algoritmus 2.107 v [1]).

I.2.4 IDEA testovanie

Na testovanie implementovaného algoritmu použijeme testovacie vektory získané z [1]. Ďalším spôsobom testovania môže byť testovanie s náhodne generovaným kľúčom a vstupom. Vstup sa zašifruje a následne spätne dešifruje a porovná sa či nastala zhoda. Ak áno, algoritmus je pre tieto vstupy funkčný. Ak nie, algoritmus je zle implementovaný.

Počítame s obidvoma spôsobmi testovania, pričom pri náhodne generovaných vstupoch chceme testovať algoritmus opakovane, teda pre viacero náhodne generovaných vstupov (zhruba stotisíc opakovaní).

Testovacie vektory pre šifrovanie:

128-bit $K = (1,2,3,4,5,6,7,8)$							64-bit $M = (0,1,2,3)$			
r	$K_1^{(r)}$	$K_2^{(r)}$	$K_3^{(r)}$	$K_4^{(r)}$	$K_5^{(r)}$	$K_6^{(r)}$	X_1	X_2	X_3	X_4
1	0001	0002	0003	0004	0005	0006	00f0	00f5	010a	0105
2	0007	0008	0400	0600	0800	0a00	222f	21b5	f45e	e959
3	0c00	0e00	1000	0200	0010	0014	0f86	39be	8ee8	1173
4	0018	001c	0020	0004	0008	000c	57df	ac58	c65b	ba4d
5	2800	3000	3800	4000	0800	1000	8e81	ba9c	f77f	3a4a
6	1800	2000	0070	0080	0010	0020	6942	9409	e21b	1c64
7	0030	0040	0050	0060	0000	2000	99d0	c7f6	5331	620e
8	4000	6000	8000	a000	c000	e001	0a24	0098	ec6b	4925
9	0080	00c0	0100	0140	—	—	11fb	ed2b	0198	6de5

obr. č.I.3 IDEA testovacie vektory pre šifrovanie

Všetky údaje v tabuľke sú v *hexadecimálnom* (číselná sústava zo základom 16) tvare veľkosti 16 bitov (štyri znaky *hexa*). Vstupom je 128 bitový kľúč K = (1,2,3,4,5,6,7,8) a vstupný 64 bitový text M = (0,1,2,3).

Sú zobrazené všetky šifrovacie kľúče a výstup pre každé kolo algoritmu a takisto po výstupnej transformácii (r=9), čo je už zašifrovaný blok.

Testovacie vektory pre dešifrovanie:

$K = (1,2,3,4,5,6,7,8)$							$C = (11fb,ed2b,0198,6de5)$			
r	$K'^{(r)}_1$	$K'^{(r)}_2$	$K'^{(r)}_3$	$K'^{(r)}_4$	$K'^{(r)}_5$	$K'^{(r)}_6$	X_1	X_2	X_3	X_4
1	fe01	ff40	ff00	659a	c000	e001	d98d	d331	27f6	82b8
2	fffd	8000	a000	cccc	0000	2000	bc4d	e26b	9449	a576
3	a556	ffb0	ffc0	52ab	0010	0020	0aa4	f7ef	da9c	24e3
4	554b	ff90	e000	fe01	0800	1000	ca46	fe5b	dc58	116d
5	332d	c800	d000	fffd	0008	000c	748f	8f08	39da	45cc
6	4aab	ffe0	ffe4	c001	0010	0014	3266	045e	2fb5	b02e
7	aa96	f000	f200	ff81	0800	0a00	0690	050a	00fd	1dfa
8	4925	fc00	fff8	552b	0005	0006	0000	0005	0003	000c
9	0001	fffe	fffd	c001	—	—	0000	0001	0002	0003

obr. č.I.3 IDEA testovacie vektory pre dešifrovanie

Aj v tejto tabuľke sú všetky údaje v *hexadecimálnom* tvare veľkosti 16 bitov (štyri znaky *hexa*). Vstupom je 128 bitový kľúč K = (1,2,3,4,5,6,7,8) a vstupný 64 bitový zašifrovaný blok C = (11fb,ed2b,0198,6de5).

Sú zobrazené všetky dešifrovacie kľúče a výstup pre každé kolo algoritmu a takisto po výstupnej transformácii (r=9), čo je vlastne dešifrovaný text.

I.3 Špecifikácia riešenia

I.3.1 Špecifikácia požiadaviek interaktívnej prezentácie

Požiadavky na interaktívnu prezentáciu sme konzultovali s vedúcim tímového projektu doc. Ing. Ladislavom Hudecom.

Interaktívna prezentácia má slúžiť na to, aby laik oboznámený so základmi logických systémov pochopil systém šifrovania pomocou šifry IDEA. Ide o interaktívnu prezentáciu, teda používateľ si má možnosť zadať svoje dáta a šifrovací kľúč. Dáta sa zadávajú buď ako nešifrované a program sa použije na ich zašifrovanie, alebo zadá šifrované dáta a tieto budú dešifrované.

Pri prezentovaní konkrétnych hodnôt sa vždy jedná o dáta vypočítané z vstupu od používateľa.

Celý postup šifrovania spolu s vysvetlením matematicko-logických funkcií bude umiestnený na tímovej web stránke spolu s interaktívnou prezentáciou.

I.3.2 Špecifikácia požiadaviek šifrátora IDEA

Šifrovací algoritmus IDEA treba opísať pomocou jazyka VHDL, aby bolo možné syntetizovať ho do štruktúry PLD obvodu. Pri návrhu bude zohľadnená veľkosť najväčšieho PLD obvodu podporovaného vývojovým prostriedkom Web Pack Xilinx [7].

Hlavným kritérium návrhu je optimalizácia na najväčšiu možnú rýchlosť. Obvod by mal byť navrhnutý tak, aby dosahoval maximálnu dátovú priepustnosť šifrovania. Na overenie funkčnosti návrhu je potrebné spraviť funkčnú simuláciu v simulátore jazyka VHLD a časovú simuláciu výstupného kódu z programového prostredia Xilinx Foundation Technology [7].

I.4 Návrh riešenia interaktívnej prezentácie

I.4.1 Návrh interaktívnej prezentácie

Prezentácia bude obsahovať tri vrstvy. Na najvyššej vrstve budú zobrazené vstupné dáta, šifrovací kľúč a výstupné dáta, všetko v *hexadecimálnom* tvare. Šifrovací kľúč má veľkosť 128 bitov. Vstupné alebo výstupné dáta majú veľkosť 64 bitov. Označením objektu znázorňujúceho šifrátor sa vyvolá zobrazenie nižšej úrovne.

Na zobrazení druhej vrstvy sa nachádzajú bloky zobrazujúce jednotlivé kolá šifrátora, spolu s vstupmi a výstupmi. Pre sprehľadnenie nebudú hodnoty vstupov a výstupov zobrazené okamžite, ale až po zaslaní požiadavky od používateľa. Po vybraní konkrétneho objektu bloku sa blok šifrátora zobrazí v zobrazení najnižšej úrovne.

Na najnižšej, poslednej úrovni, sú zobrazené všetky matematicko-logické funkcie a ich prepojenie. Opäť nie je vhodné zobrazovať všetky hodnoty na obrazovke implicitne, ale až po požiadavke od používateľa. Pri logických funkciách budú hodnoty zobrazované v binárnej číselnej sústave aby si užívateľ ľahšie mohol daný výsledok prepočítať sám.

I.4.2 Výber implementačného prostredia interaktívnej prezentácie

Interaktívna prezentácia šifry IDEA sa má nachádzať na tímovej web stránka. Z tohto pohľadu sa treba pozerať aj na problém výberu implementačného prostredia. Dajú sa rozdeliť na prostriedky, ktoré sa spúšťajú na web serveri a prostriedky spúšťané na klientskom počítači.

Na výber sme mali programovacie prostriedky: *PHP, Java aplet, Java*

script a Flash. Rozhodli sme sa pre prostriedok *Flash* od firmy *Macromedia* [8]. Pre tento prostriedok sme sa rozhodli preto, lebo sa najviac hodí na tvorbu interaktívnych prezentácií a je možné spúšťať prezentáciu aj z iného média ako je Internet alebo pracovať s prezentáciou bez pripojenia na Internet. Nevýhodou tohto riešenia je potreba pomocného modulu do prehliadača.

I.4.3 Návrh grafickej časti prezentácie šifry IDEA

Popri prezentácii samotného algoritmu práce šifry IDEA je dôležitý aj slovný opis činnosti a vysvetlenie prípadných odborných pojmov. Rovnako sa snažíme o vytvorenie prehľadného a intuitívneho prostredia. Za týmto účelom môžu byť jednotlivé úrovne odlíšené použitím rozdielnych farieb a hlavné okno prezentácie bude rozdelené na tri časti.

- V hornej časti sa pre jednoduchšiu orientáciu bude nachádzať okno informujúce o prezentačnej vrstve v ktorej sa práve nachádzame, môžu sa tu nachádzať aj odkazy na zvyšné vrstvy.
- V hlavnom okne bude graficky znázornená činnosť šifry na zvolenej úrovni. Všetky funkčné bloky diagramu musia byť pomenované rovnako ako vstupné a výstupné polia hodnôt.
- V pravej časti prezentácie bude okno v ktorom bude zobrazované doplňujúce informácie o obsahu hlavného okna a vysvetlenie použitých pojmov z oblasti šifrovania. Tieto údaje sa budú zobrazovať po *kliknutí* na aktívnu časť, alebo odkaz v hlavnom okne.

Pri prechode medzi jednotlivými úrovňami je potrebné zachovanie hodnôt vstupu, výstupu a prípadných hodnôt medzi jednotlivými operáciami, ktoré môžu byť zobrazené formou bublinkovej nápovedy *(text zobrazený v okienku tesne nad ukazovateľom myši)* pri pohybe myšou cez dátový prepoj, alebo implicitne v informačnom okne. Tu treba pri impementácii algoritmu šifry počítať s potrebou udržiavať všetky hodnoty dočasných výpočtov, ktoré chceme mať prístupné v prezentácii.

Navrhované časti prezentácie:

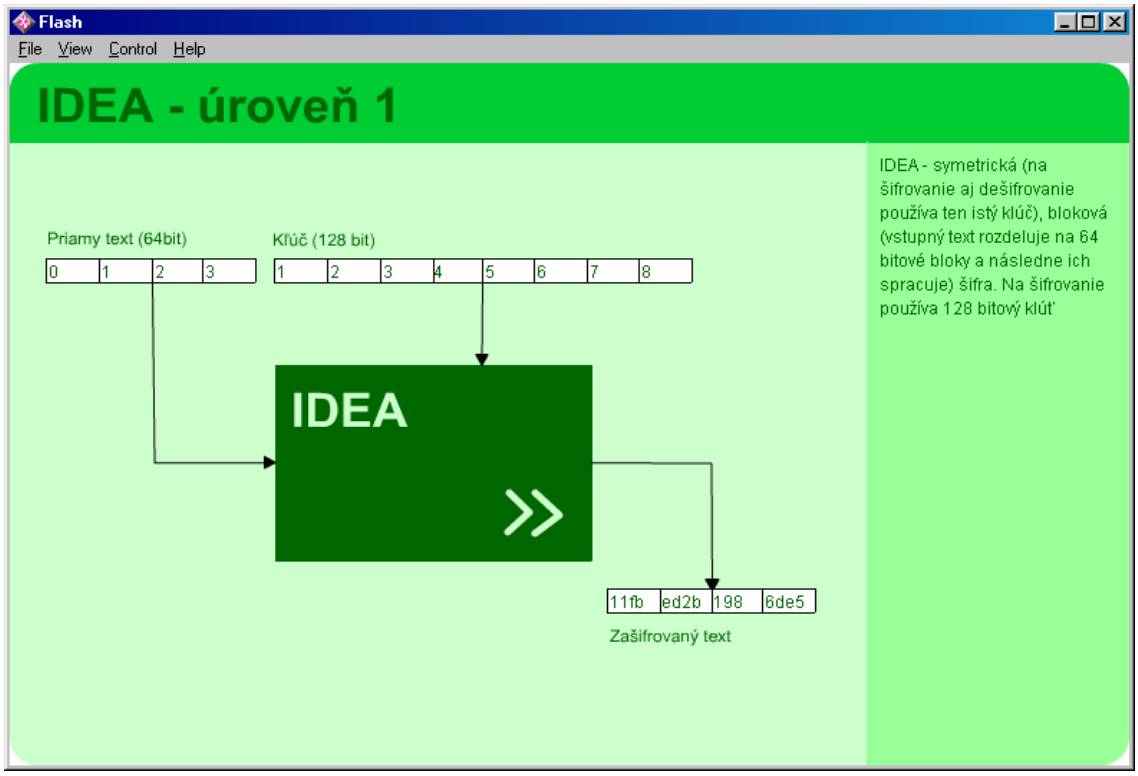
- Najvyššia úroveň zobrazuje šifru ako čiernu skrinku zo vstupnými údajmi priamy text šifrovací kľúč a výstupným zašifrovaným textom
- Úroveň jednotlivých kôl šifrovania so znázornenými dátovými tokmi medzi jednotlivými kolami. Zobrazených bude 8 kôl a blok spracovania šifrovacieho kľúča.
- Úroveň jedného kola so znázornenými bitovými operáciami. Toto zobrazenie bude spoločné pre prvých sedem kôl.
- Úroveň posledného kola na bitovej úrovne do ktorého vstupujú iba štyri vygenerované kľúče na rozdiel od predchádzajúcich kôl do ktorých vstupuje šesť kľúčov.
- Spracovanie šifrovacieho kľúča na bitovej úrovni.

I.5 Implementácia

I.5.1 Implementácia grafického rozhrania prezentácie

Na tvorbu prezentácie sme sa rozhodli využiť nástroj *Flash* 5 firmy Macromedia, ktorá umožňuje tvorbu interaktívnych webových stránok a jednoduchých animácií [5] [6] [8]. Na prezeranie výsledného produktu je potrebný prehliadač www stránok s inštalovaným modulom prehrávača *Flash* animácií, ktorý je volne stiahnuteľný na Internete.

V súčasnej dobe sú rozpracované prvé dve úrovne prezentácie šifry. Obrázok č.I.4 zobrazuje najvyššiu úroveň. Okno je rozdelené na tri časti ktoré zobrazujú aktuálnu úroveň, náhľad na šifru z najvyššej úrovne a časť zobrazujúcu spresnený popis zvolených častí v hlavnom okne.



Obr. č.I.4 Najvyššia úroveň prezentácie šifry IDEA.

I.5.2 Implementácia algoritmu prezentácie

Pri implementácii sme vychádzali z opisu šifrovacieho algoritmu v jazyku C, ktorý je možné nájsť v [3]. Celý algoritmus bol implementovaný v jazyku *„Action Script“* [5], ktorý je súčasťou zvoleného implementačného prostredia *Macromedia Flash*.

Pri vykonaní algoritmu sa budú uchovávať nasledovné údaje:

- Používateľom zadaný vstupný text a kľúč
- Všetky vygenerované šifrovacie aj dešifrovacie kľúče
- Vstupy a výstupy každého kola algoritmu
- V rámci každého kola algoritmu všetky vstupy a výstupy každého bloku (XOR, sčítanie násobenie)
- Všetky údaje budú v základnom tvare a pri vizualizácii sa podľa potreby budú transformovať do príslušnej sústavy (binárna, šestnástková, desiatková)

Vizualizačná časť bude zabezpečovať zobrazenie iba tých údajov, o ktoré bude mať používateľ záujem. Cieľom je nezahltiť používateľa nepotrebnými údajmi, ktoré by pôsobili neprehľadne.

I.6 Literatúra

[1] A. Menezes, P. van Oorschot, and S. Vanstone: Handbook of Applied Cryptography, CRC Press, 1996.

[2] A. Buldas, J. Poldre: A VLSI Implementation of RSA and IDEA encryption engine

[3] Fauzan Mirza: International Data Encryption Algorithm – Implementation summary

[4] Encryption solutions for secure communication, http://www.mediacrypt.com, (október 2002)

[5] Macromedia Inc. 2000. Macromedia Flash 5 Action Script.

[6] Macromedia Inc. 2000. Macromedia Flash 5 Užívateľská príručka.

[7] XILINX homepage, http://www.xilinx.com.

[8] Macromedia homepage, http://www.macromedia.com