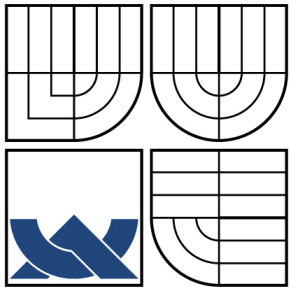
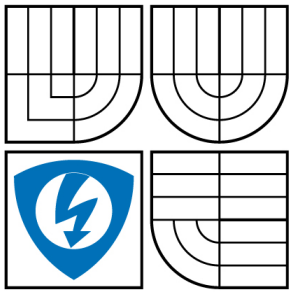


VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

INTERAKTIVNÍ APLIKACE PRO DVB-MHP

DVB-MHP INTERACTIVE APPLICATION

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

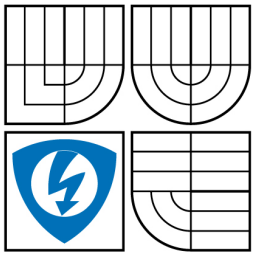
AUTOR PRÁCE — Bc. PETR KASAL

AUTHOR

VEDOUCÍ PRÁCE — Ing. PETR ČÍKA

SUPERVISOR

BRNO 2009



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Petr Kasal	***ID:***	84318
Ročník:	2	***Akademický rok:***	2008/2009

NÁZEV TÉMATU:

Interaktivní aplikace pro DVB-MHP

POKYNY PRO VYPRACOVÁNÍ:

Seznamte se s platformou MHP pro vytváření interaktivních aplikací pro DVB. Navrhněte a realizujte aplikaci umožňující vyplňování anket a jejich archivace na vzdáleném serveru. Komunikace MHP aplikace se serverem bude probíhat po zabezpečeném kanále. Základem celé aplikace bude uživatelsky přístupné grafické rozhraní.

DOPORUČENÁ LITERATURA:

[1] Schwalb, E. M. ITV Handbook: Technologies and Standarts, London: Prentice Hall, 2004. ISBN 978-0131003125
[2] STEVEN, M. Interactive TV standards: A Guide to MHP, OCAP, and JavaTV. Elsevier: Focal Press. 2005, ISBN 978-0240806662

Termín zadání:	9.2.2009	***Termín odevzdání:***	26.5.2009

Vedoucí práce: Ing. Petr Číka

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

Abstrakt

Tato práce se zabývá standardem interaktivní televize. V první části je nastíněna obecná problematika digitální televize a její přínos. Následující kapitola pojednává o interaktivitě pro systémy digitální televize. Dále jsou nastíněny možnosti zabezpečení a ochrany dat při přenosu přes zpětný kanál interaktivní televize. Hlavní část práce se věnuje vývoji aplikace umožňující získávání statistických dat od diváků prostřednictvím set-top boxu. Aplikace je pojata jako univerzální anketa, jenž si načítá aktuální data pomocí zpětného kanálu z internetu a přehlednou formou se snaží prezentovat získané údaje. Je použito programovacího jazyka Java a PHP a data jsou uložena v MySQL databázi s možností snadné editace. Při komunikaci je využíváno vlastních zabezpečovacích knihoven vycházejících z šifry XTEA.

Klíčová slova

digitální televize, MHP, šifrování, XTEA, Java, PHP

Abstract

This Master Thesis is engaged in Interactive TV standard. In the first chapter a general digital TV question and its benefits are adumbrated. Next chapter deals with Interactivity for digital TV itself. More on possibilities of data security and protection during the transmission throught the reversal channel are discussed. The main part of the Thesis deals with developing an application enabling to obtain a statistical data from televiewers throught a set-top box connected to the Internet. The application is conceived as an universal enquiry, that obtains its data via a reversal channel and is able to present the gained data in an attractive way. Java and PHP programming languages were applied to create the application and MySQL database with easy edit system is used to store the needed data. Specially designed security libraries based on XTEA cipher are used to enable protected communication.

Key words

digital television, MHP, encryption, XTEA, Java, PHP

KASAL, P. *Interaktivní aplikace pro DVB-MHP*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2009. 64 s. Vedoucí diplomové práce Ing. Petr Číka.

Prohlášení o původnosti práce

Prohlašuji, že svou diplomovou práci na téma „Interaktivní aplikace pro DVB-MHP“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne …… ……

(podpis autora)

1. Obsah

2. Úvod do problematiky

Stávající analogové rozhlasové a televizní vysílání, které má v naší republice více než padesátiletou tradici, je v posledních letech konfrontováno s rozmáhajícím se vysíláním digitálním, které oproti klasickému analogovému přenosu nabízí mnoho dalších možností využití a zajímavých technických vlastností. Střet těchto technologií spočívá v tom, že oba systémy pro své šíření využívají stejná kmitočtová pásma, neboť jiná již nejsou z technických důvodů dostupná. Proto je nutné řešit, jak zabezpečit prostor pro nové digitální vysílání v pásmech již obsazených analogovým provozem.

Proces přechodu z dosavadního zemského analogového televizního způsobu šíření signálu na zemské digitální vysílání je v současné době již prakticky odstartován a probíhá nejen v ČR, ale v celé Evropě, samozřejmě s různou intenzitou v závislosti na konkrétní situaci v té které zemi.

Úvod této práce se ve stručnosti věnuje základním vlastnostem digitálního vysílání, jeho výhodám oproti analogovému a jednotlivým typům přenosu televiznímu signálu prostředím, zejména DVB-T. Následující část je věnovaná interaktivnosti; principům fungování a možným způsobům uplatnění interaktivní platformy MHP, způsobu realizace interaktivních aplikací na platformě Java-MHP, a jejím teoretickým základům. Další kapitola je věnována možnostem zabezpečení dat z hlediska autentičnosti a především důvěrnosti. Hlavní část práce pojednává o konkrétním řešení aplikace na platformě MHP, její tvorbě, odlaďování a spouštění.

3. Digitální televizní vysílání

3.1 Výhody přechodu na digitální vysílání

Digitální způsob televizního vysílání nabízí celou řadu nových možností [1]. Jedná se především o:

- Větší nabídka televizních a rozhlasových kanálů, nabídka moderních doplňkových interaktivních služeb – namísto jednoho analogového televizního programu můžeme v daném prostoru vysílat až pět TV programů přenášených v digitální podobě, sedm rádiových stanic a balík interaktivních služeb na bázi MHP. To je umožněno hlavně komprimací datového toku.
- Uvolnění vysílacího prostoru vede ke vstupu dalších subjektů provozujících digitální vysílání, což příznivě ovlivní rozvoj konkurenčního prostředí.
- Signál je dostupnější i v členitých, např. horských oblastech.
- Lepší využití frekvenčního spektra, redukce emisí elektromagnetického vysílání a úspora energie.

Charakteristická pozitiva digitálního vysílání:

- vysílače v síti mohou pracovat na stejné frekvenci – síť typu SFN
- kontrolní a korekční kódy
- úroveň a kvalita signálu jsou do jisté míry zjistitelné a monitorovatelné

3.2 Moderní interaktivní a informační služby

Velice zajímavou vlastností digitálního přenosu je rozšiřitelnost televizních a rozhlasových pořadů o řadu dalších doplňkových datových služeb, mezi které patří i zpětná interaktivita při sledování digitálního vysílání, což je v televizním vysílání unikát. Tyto služby do budoucna skýtají obrovský potenciál a mnoho způsobů uplatnění; záleží už jen na zájmu diváků o tyto služby, jejich provozně právním zakotvení v zákonech a na samotné technické realizaci přístrojů a technologií umožňujících jejich využití.

3.3 Základní členění digitálního televizního vysílání

Podle způsobu transportu televizního signálu můžeme rozlišit tyto kategorie digitálního vysílání [5, 8]:

- **Satelitní vysílání** (DVB-S – *Digital Video Broadcasting-Satelite*) Tento standard je používán Evropě a ve větší části světa. Transportní kanál je velmi chybový, s velkým útlumem a velkou vzdáleností mezi vysílačem a přijímačem.

- **Vysílání prostřednictvím kabelové televize** (DVB-C – *Digital Video Broadcasting - Cable*) Typ použitého média u kabelové televize je charakteristický především svou nízkou úrovní rušení; přenosový kabelový kanál se svými vlastnostmi blíží ideálnímu transportnímu kanálu.

- **Pozemní vysílání pro přenosná zařízení** (DVB-H – *Digital Video Broadcasting-Handhelds*) Tento systém je realizován obdobně jako technologie DVB-T, signál se na mobilní zařízení šíří z běžných televizních vysílačů.

- **Vysílání přes internetové médium na IP protokolu** (IPTV – *Internet Protocol Television*) Toto vysílání pro svůj přenos využívá širokopásmových internetových přípojek, nejčastěji ADSL, ADSL2+.

  Kanál je ze své podstaty duplexní, což je velká výhoda, jenž umožňuje existenci plnohodnotné zpětné vazby mezi příjemcem a vysílatelem. Tato vlastnost je velmi vítaná právě z hlediska interaktivity.

- **Pozemní vysílání** (DVB-T – *Digital Video Broadcasting-Terrestrial)* Viz kapitola 3.3.1.

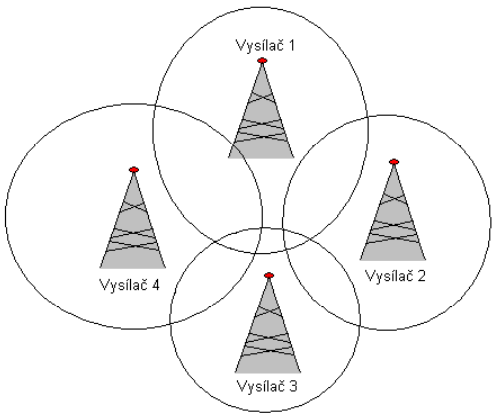
3.3.1 Pozemní vysílání DVB-T

Jedná se o šíření televizního signálu pomocí sítě televizních vysílačů. Přenosovým médiem je u tohoto druhu transportu dat volné prostředí, z čehož pramení, že musíme řešit problémy jako např. nelineární zkreslení, kolísání útlumu přenosového kanálu, labilnost vůči šumu a rušivým signálům, ale zásadní je mnohočetnost šíření a častý výskyt odrazů signálu.

Při procesu digitalizace dat dochází k velkému nárůstu irelevance a redundance dat, proto musíme aplikovat kódy snižující datový tok. Pro DVB-T využíváme kompresních metod MPEG2, výjimečně MPEG4 AVC.

Pozemní vysílání zaujímá I. až V. vysílací pásmo, což odpovídá kmitočtu 47 až 800 MHz. Pro DVB-T je používána OFDM modulace (*Orthogonal Frequency Division Multiplexing* – Ortogonální Frekvenčně Dělený Multiplex) a zřetězená digitální modulace COFDM (*Coded Orthogonal Frequency Division Multiplexing*)

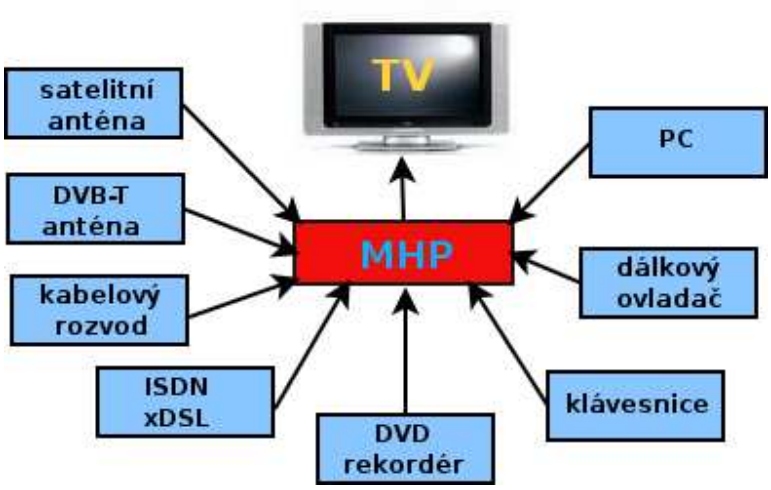
Počítá se s využitím sítě vysílačů, které budou pracovat na stejném kmitočtu (SFN = *Single Frequency Network*) a příspěvky z jednotlivých sítí se budou sčítat ve prospěch výsledné kvality signálu. Maximální velikost takovéto buňky několika vysílačů je až 67,2 km (vychází z omezení použité technologie OFDM) [1].



Obrázek 3.1: Jednofrekvenční síť (SFN – S*ingle Frequency Network*).

4. Interaktivní platforma MHP digitálních televizních systémů DVB

Při návrhu vlastností platformy digitálního vysílání DVB byl jeden z požadavků obecně kompatibilní systém divácké interaktivity, umožňující oboustranný průtok informací, jenž by mohl aktivně kooperovat s koncovým uživatelem. Z tohoto důvodu vznikla platforma MHP (*Multimedia Home Platform*) a při rozvoji digitální televize se s ní počítá do jisté míry jako závaznou jak pro distributory signálu, tak pro výrobce hardware. Mluvíme-li o MHP, pak se jedná o celoevropský systém, jenž je otevřený a jenž by měli využívat všichni tvůrci multimediálních domácích zařízení pro digitální televizní vysílání. MHP přináší vylepšení zobrazení, např. zvolení úhlu pohledu, výběr kamery, ale hlavním pilířem je interaktivita a posléze i konektivita na internet.

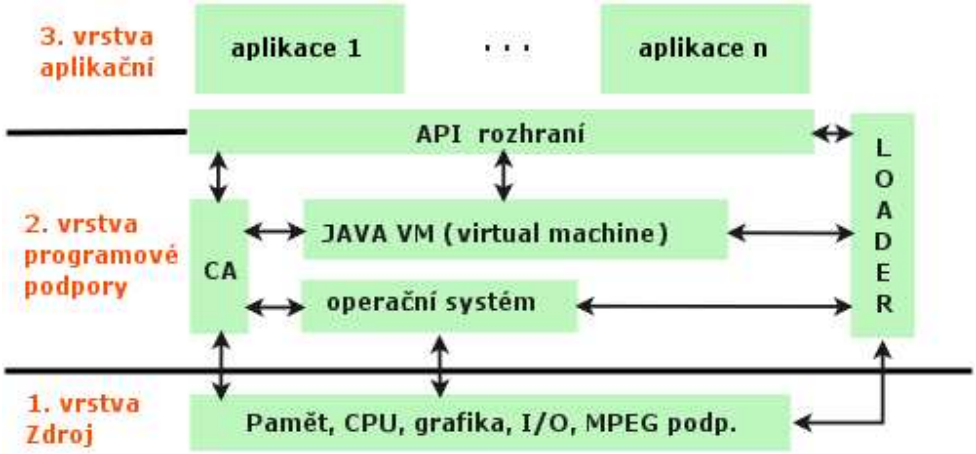


Obrázek 4.1: MHP a její propojení (rozhraní) navenek.

4.1 Základní struktura MHP platformy

U platformy MHP můžeme hovořit o třech základních vrstvách:

- Zdroj
- Programová podpora
- Samotný program (aplikace)



Obrázek 4.2: Struktura vrstev MHP.

V MHP není definován hardware jako nejnižší vrstva. Je na výrobcích zařízení, jak tuto vrstvu implementují. Jsou zde obsaženy až vrstvy vyšší.

Zdroj zpravidla obsahuje procesor, paměti, grafickou podporu a podporu pro práci se standardem MPEG.

Programová podpora je další vrstva umožňující aplikacím správný chod, spouštění a ukončování a přístup k prostředkům nižších vrstev. Dalo by se říct, že se jedná o operační systém spravující požadavky vyšších vrstev přerozdělováním zdrojů vrstvy nižší. Důležitou součástí je aplikační manažer, pomocí něhož se jednotlivé aplikace spouští, řídí a ukončují. Pro aktualizaci software obsahuje vrstva tzv. „*Loader*“, jenž je implementován dle výrobce. Důležitou součástí této vrstvy je JAVA VM (*JAVA Virtual Machine*), což je de facto zapouzdření hardware pro využití aplikační vrstvou. To znamená, že žádá-li aplikace přístup k hardwarovým zdrojům, činí tak přes JAVA VM a sama nevidí co je pod touto vrstvou obsaženo.

Do aplikační vrstvy spadají jednotlivé programy, řízené aplikačním manažerem druhé vrstvy. Tyto aplikace se spouštějí nad vrstvou aplikačního rozhraní API (*application programming interface*), která má pevně definovaná pravidla přístupu a proto umožňuje oddělit samotnou aplikaci od nižších vrstev a dosáhnout tak multiplatformnosti [3].

4.2 Interaktivita z pohledu existence zpětného kanálu

4.2.1 Interaktivita bez zpětného kanálu

Interaktivita bez zpětného kanálu je aplikací vysílanou souběžně s hlavními obrazovými a zvukovými daty. Sama pro svůj chod nevyžaduje přítomnost zpětného kanálu. Jedná se například o službu teletext, kde jsou všechna data přenášena až k divákovi a ten pouze prostřednictvím ovladače volí, jaké informace chce zobrazit.

Na příkladu teletextu je vidět celou myšlenku prvního typu interaktivity. Všechny ostatní aplikace s tímto typem interaktivity pouze rozvíjí tento základní přístup. Prostor k vylepšování je hlavně ve volbě přenášených obrázků, ozvučení, grafiky, pohodlnosti a intuitivnosti ovládání, rychlosti odezvy atd. Podobných aplikací je v Evropě už celá řada, např. španělská aplikace předpovědi počasí nabízí vyspělý systém informací o počasí s množstvím různých voleb zobrazení, výběru zemí, velice zdařilého grafického zpracování a uživatelky přívětivého prostředí s uložením konfigurace pro další načtení aplikace. Tento druh interaktivity je vlastně předstupněm interaktivity v pravém slova smyslu. Neměli bychom však opomíjet ani tuto částečnou interaktivitu (bez zpětného kanálu), a to zejména kvůli menší hardwarové náročnosti. Zejména má-li se jednat o celostátně využívanou platformu je to velice důležitý aspekt. Vezmeme-li v potaz velikost kupní síly obyvatelstva, je daný systém velice lukrativním předstupněm příchodu interaktivity se zpětným kanálem a rozhodně stojí za rozvíjení a zahrnutí do vysílací nabídky.

4.2.2 Interaktivita se zpětným kanálem

Úplnou interaktivitou rozumíme, je-li divák připojen do sítě, nejčastěji do sítě internet, za pomoci dalšího komunikačního kanálu, tzv. zpětného kanálu, jenž je oboustranně průchozí. Tímto kanálem může být standardní telefonní síť, přístup ADSL, či mobilní sítě typu GSM nebo UMTS.

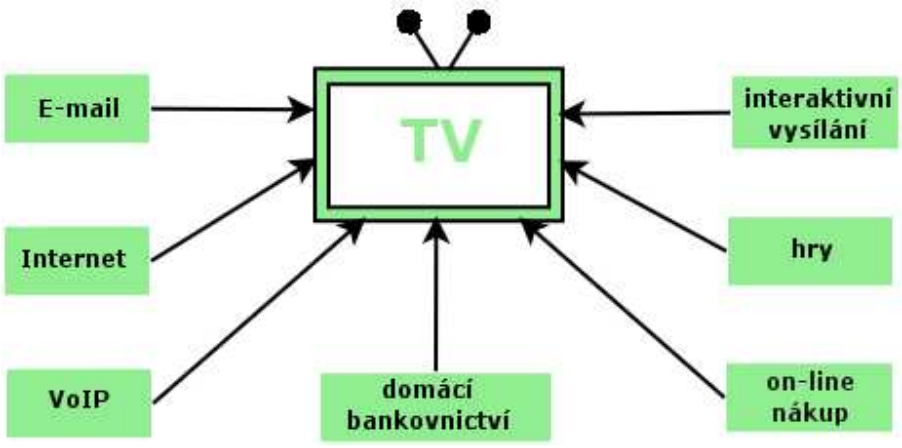
Jde o podobný princip, jako když používáme SMS hlasování, kde divák rozhoduje prostřednictvím zaslání krátké textové zprávy SMS obsahující příslušný kód na číslo mobilního operátora. Avšak v MHP přijímačích s podporou zpětného kanálu bude odeslání SMS zprávy buďto plně automatizováno, nebo bude dokonce využito jiného přenosového systému (uživatel už nebude nucen psát žádné SMS a jeho preference se automaticky odešlou zpět do vyhodnocovacího centra). Není ještě jisté, jaká technologie pro zpětný kanál převládne. Ať už se bude jednat o jakoukoli technologii, idea úplné interaktivity zůstává zachována. Divák se posouvá z pozice pasivního konzumenta blíže k oblasti aktivního tvůrce programové náplně televizního vysílání.

Do budoucna se počítá s využitím interaktivity se zpětným kanálem i pro daleko sofistikovanější technologie využívající bezpečné spojení s jejich provozovatelem, které umožní např. objednání zboží prostřednictvím televize, přístup k bankovnictví, poště a internetovým aplikacím a to přes televizní obrazovku [1].

4.3 Požití platformy MHP:

Užití interaktivní multimediální platformy pro potřeby domácností je obrovské. Ať už se jedná o vylepšeného programového průvodce (EPG), nákupy na internetu, video na požádání, sázky, ankety, průzkumy a mnoho dalšího [4]. Jde pouze o způsob včasného a dostatečného rozšíření těchto aplikací do domácností, jenž je podmíněno dostatečnou přitažlivostí nabídky služeb a mírou dostupnosti pro koncového uživatele. Z nejvíce atraktivních možností nabízejících se divákovy jmenujme tyto [1]:

- **Home shopping** – nakupování z domova
- **Home banking** – provádění bankovního styku z domova
- PPV = **Pay-per-view** – systém placení jen za zhlédnuté pořady
- VoD - **Video on Demand** – divák si volí film dle svého uvážení
- AoD - **Audio on Demand** – divák si volí hudbu dle svého uvážení
- **interaktivní vysílání televize** – soutěžní pořady, hry o ceny, průzkumy názorů
- **E-mail, přístup na internet**
- VoIP - **Voice over IP** - hlas přenášený internetem



Obrázek 4.3: Příklad služeb poskytovaných MHP aplikacemi.

4.4 Vývoj standardu interaktivních služeb

V důsledku toho, že se dlouhou dobu čekalo na příchod mezinárodně vytvořeného a uznávaného standardu, který však dlouhou dobu nepřicházel, rozhodli se někteří výrobci pro vývoj svých vlastních platforem. Tyto platformy však byly navzájem nekompatibilní. Jednalo se např. o *Open TV*, *Beta Nova*, *Media Highway*. Toto způsobilo rozkouskování trhu a každý divák, který chtěl využívat interaktivních aplikací od různých poskytovatelů, byl nucen využívat různého hardwaru, nebo se spokojit s přístupem k omezenému počtu aplikací.

Aby byla filosofie interaktivních služeb dostupných každé domácnosti realizovatelná, bylo potřeba sjednotit trh a vytvořit celoevropsky uznávaný standard multimediálních interaktivních aplikací, který by byl závazný pro všechny výrobce a distributory. Experti na DVB problematiku se rozhodli vytvořit multimediální domácí platformu, kterou nazvali MHP. Tato platforma stojí na API rozhraní (*Application Programming Interface*). Standard MHP postihuje problematiku jako celek, a to od vzniku datového streamu nesoucího interaktivní aplikace, přes příjem na domácích zařízeních pro digitální televizní příjem (set-top boxech) až pro propojení jednotlivých zařízení v domácnosti do jedné lokální domácí sítě, tzv. LHN (*Local Home Network*).

Vývoj tohoto interaktivního standardu vycházel ze série požadavků, které měli být splněny, aby mohla být nová multimediální platforma aplikovatelná do praxe. Experti na DVB vypracovali seznam bodů, vyžadovaných při vývoji této platformy, které byly konzultovány s výrobci multimediálních zařízení, a to tak, aby byla dosažena co největší komerční výhodnost a technologická jednota [5].

Mezi hlavní požadavky kladené na nově vznikající standard patřily tyto:

- **Otevřený a volně modifikovatelný standard.**
- **Interoperabilita a kooperativnost platforem** odlišných výrobců. Aplikace musí být spustitelné nehledě na výrobce hardwaru či druhu providera.
- **Zabezpečení** – platforma MHP musí být bezpečná proti virům, musí umožňovat ochranu proti změnám a napadení dat, ochranu autorských práv, ochranu zpětného kanálu proti zneužití třetí stranou, ochranu osobních dat uživatele.
- **Vrstvový model** MHP platformy**.**
- **Schopnost upgradu** a to přímou cestou nebo přes zpětný kanál**.**
- **Umožnění rozšiřitelnosti standardu** do budoucna.
- **Intuitivní obsluha aplikací**, využívat ji bude zejména laická veřejnost.
- **Minimální finanční zatížení** pro spuštění interaktivních služeb.
- **Nezávislost API a CA** (*Conditional Access System* – jednotka podmíněného přístupu), musí být umožněna podpora odlišných CA přístupů a Common Interface [1].

Při volbě struktury a způsobu vytváření interaktivních aplikací se jako nejslibnější systémy ukázali dva, a to aplikace psané jazykem HTML (*Hypertext Markup Language*) a systém využívající jazyk Java. HTML jazyk je textový a využívá se hlavně na stránkách internetu. Java je programovací jazyk vyvinutý firmou Sun a taktéž je koncipován pro tvorbu internetových aplikací. Z těchto dvou jazyků byla vybrána druhá varianta, a proto se MHP aplikace zakládají na tvorbě ve vývojovém prostředí Java, jež je dále přizpůsobeno potřebám interaktivních aplikací.

Jako první vznikl standard MHP 1.0. Ten byl posléze rozšířen o standardy 1.0.1 a 1.1, jež vycházely, a dále rozšiřovali možnosti MHP 1.0. Dalším krokem je standard MHP 2.0, ve kterém se už počítá s koncepcí jednotné lokální domácí sítě a propojení na ostatní spotřebiče. Co se týče HTML je snaha zaimplementovat podporu tohoto jazyka do specifikace MHP, přičemž se tak daří až od verze MHP 1.1, v níž je i specifikace pro prohlížení webových stránek a přístup na email [7].

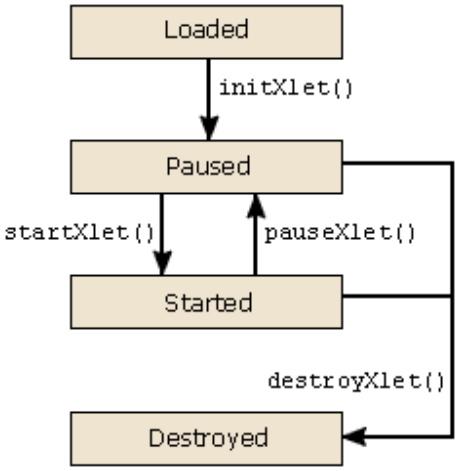
4.2 Životní cyklus MHP aplikace, Xlet

Potřeby interaktivní digitální televize nejsou zcela totožné s potřebami aplikací vyvíjenými pro běžné počítače, a proto bylo třeba některé věci upravit. Výsledkem je vznik Xletu, což je obdoba Java appletu, avšak uzpůsobená potřebám platformy MHP. Stejně jako applety tak i Xlety používají rozhraní, které povoluje externí zdroje (což má na starosti správce aplikací, který především umožňuje spuštění a ukončení dané aplikace). Název Xlet se začal používat kvůli podobnosti DVB-J s obecným Java appletem, proto se častěji než DVB-J používá název Xlet.

Hlavním rysem přístroje, jakým je set-top box, je možnost současného běhu více aplikací. Hardwarové omezení nám však dovolí vidět vždy jen jednu aplikaci v daném okamžiku, a proto ostatní aplikace musí být pozastaveny, aby zdroje byly uvolněny právě pro chod dané aplikace, která se aktuálně zobrazuje [3].

Hlavní stavy Xletu jsou:

- **načtený** (*Loaded*),
- **pozastavený** (*Paused*),
- **spuštěný** (*Started*),
- **ukončený** (*Destroyed*).



Obrázek 4.4: Stavový diagram Xlet aplikace.

4.2.1 Stav Xletu – načtený (*Loaded*)

Správce aplikací (navigátor) načte soubor obsahující hlavní třídu Xletu a vytvoří instanci Xletu zavoláním defaultního konstuktoru. Jakmile se to stane, Xlet je převeden do stavu načtený (*Loaded*).

4.2.2 Stav Xletu – pozastavený (*Paused*)

Jestliže uživatel zvolí start aplikace (případně AIT - *Application Information Table*, nebo jiná signalizační aplikace nastaví, že by Xlet měl startovat automaticky), pak správce aplikací zavolá metodu *initXlet()* a vytvoří nový objekt *XletContext* vztahující se k tomuto novému Xletu.

Xlet může použít tohoto objektu *XletContex*, aby provedl inicializaci na sobě samém, případně aby přednačetl nějaké větší celky (např. obrázky vyžadující delší čas na načtení z objektového karuselu). Jakmile je inicializace dokončena, nachází se Xlet ve stavu pozastavený (*Paused*) a je připraven ke spuštění a okamžitému použití.

4.2.3 Stav Xletu – spuštěný (*Started*)

Po obdržení návratové hodnoty vykonáním metody *initXlet()* zavolá správce aplikací další metodu a to *startXlet()*. Tímto dojde k přesunu Xletu ze stavu pozastavený do stavu spuštěný (*Started*). Nyní je Xlet připraven začít komunikovat s uživatelem.

V rámci průběhu programu se může stát, že správce aplikací zavolá metodu *pauseXlet()*. Tím přejde Xlet zpět do stavu pozastaven. Později se Xlet může vrátit zpět do stavu spuštěn zavoláním metody *startXlet()*. Toto může probíhat opakovaně, záleží na správci aplikací.

4.2.4 Stav Xletu – ukončený (*Destroyed*)

Na konci životního cyklu Xlet aplikace zavolá správce aplikací metodu *destroyXlet()*. To přesune Xlet do stavu ukončený (*Destroyed*) a uvolní všechny jeho zdroje. Po provedení tohoto kroku nemůže být daná instance Xletu již znovu aktivována.

4.5 Transportní protokoly

Služby konceptu DVB počítají s přenosem jak televizního, tak i interaktivního datového toku. Dále pro zajištění interoperability bylo nutné určit povinně implementované protokoly jak pro cestu od vysílače k divákovi, tak i od diváka zpět v distributorovi.

Protokoly dělíme [1]:

1. **protokoly pro streamed video a audio**
    - DVB Objekt Karusel
    - DSM-CC Datový Karusel
    - MPEG-2
    - MPEG-2 Transport stream

2. **protokoly pro data a aplikace**
    - TCP – protokol spolehlivého spojovaného přenosu na transportní vrstvě
    - UDP – protokol nespolehlivého nespojovaného přenosu na transportní vrstvě
    - IP – *Internet protokol*, protokol síťové vrstvy

4.6 Profily – oblasti využití MHP

Platforma MHP ve své komplexnosti obsahuje i požadavky na výrobce přijímačů včetně definice optimalizací, minimálních požadavků MHP a tří základních profilů, členících MHP dle podporovaných technologií. Z minimálně vyžadovaných prostředků můžeme zmínit rozlišení alespoň 720 x 576 pixelů, tři různé stupně průhlednosti, podporu Mpeg I-Frame, specifické fonty a podporu zpracování vstupu z klávesnice dálkového ovladače [1].

Rozlišujeme tyto profily:

1. **Vylepšené televizní vysílání (*Enhanced Broadcast*)**

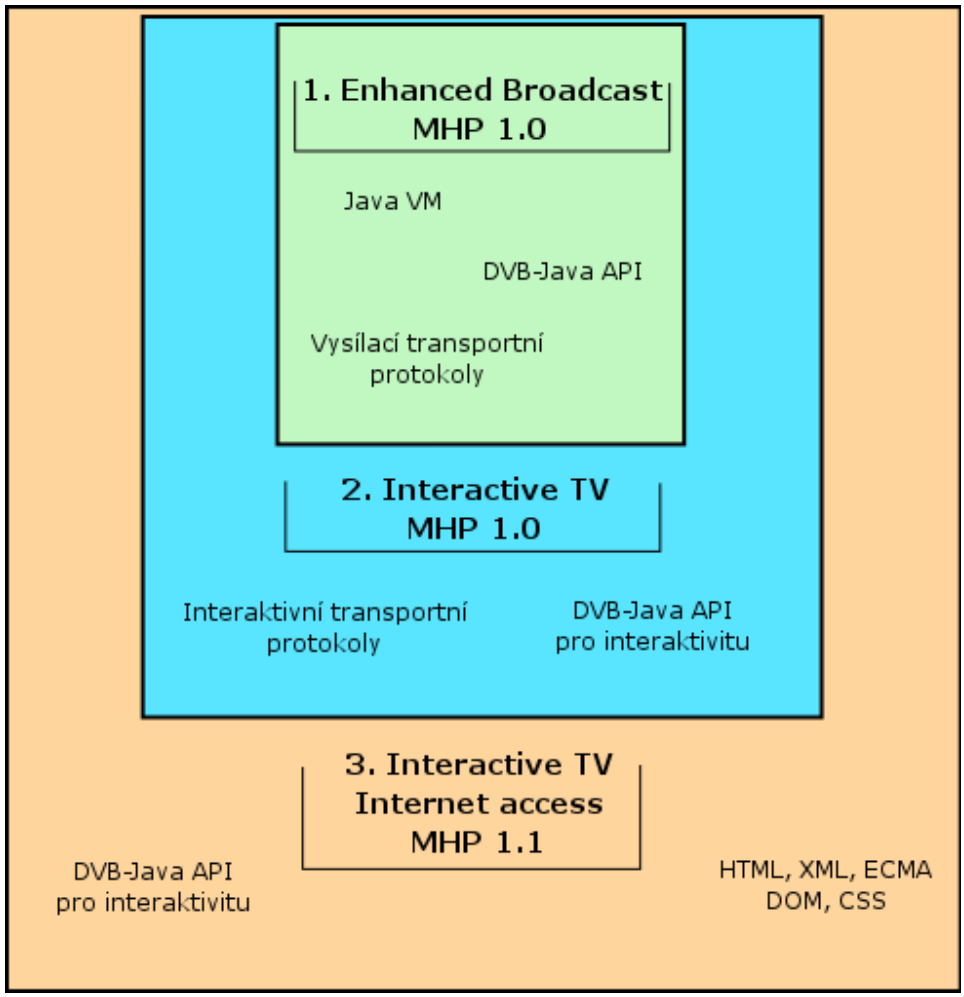
   Set-top boxy neobsahují zpětný kanál, je použita pasivní interaktivita (divák získává informace pouze přes přímou vysílací cestu). Tento profil je zakotven standardem MHP 1.0. Jeho součástí je Java VM (Virtual Machine), dále DVB-J API a vysílací transportní protokoly.

2. **Interaktivní televizní vysílání (*Interactive TV*)**

   Tento profil je zakotven taktéž standardem MHP 1.0, set-top boxy však podporují vyšší stupeň interaktivity a obsahují zpětný kanál. Součástí je i rozšířená aplikace DVB-J API pro interaktivitu a obsahuje interaktivní transportní protokoly založené na IP.

3. **Interaktivní televizní vysílání s přístupem k internetu (*Interactive TV – Internet Access*)**

   Profil vychází ze set-top boxů s vysokým hardwarovým výkonem a velkou pamětí a podporuje nejvyšší stupeň interaktivity. Je definován specifikací MHP 1.1. Jeho součástí je Java API pro přístup k internetu, dále transportní vysílací protokoly IP, podpora DVB-HTML atd.



Obrázek 4.5: Základní profily MHP aplikací.

5. Principy šifrování komunikace

Potřeba šifrování a zabezpečení dat je v dnešní době velmi aktuální a pro vývoj moderních aplikací se musí tato otázka zohlednit.

Při komunikaci po sdíleném médiu, jakým je standardní sítové internetové připojení použité pro realizaci zpětného kanálu set-top boxu, je nebezpečí, že dojde k útoku na přenášená data. Proto je v zadání k vytvářené aplikaci i realizace vlastního typu zabezpečení.

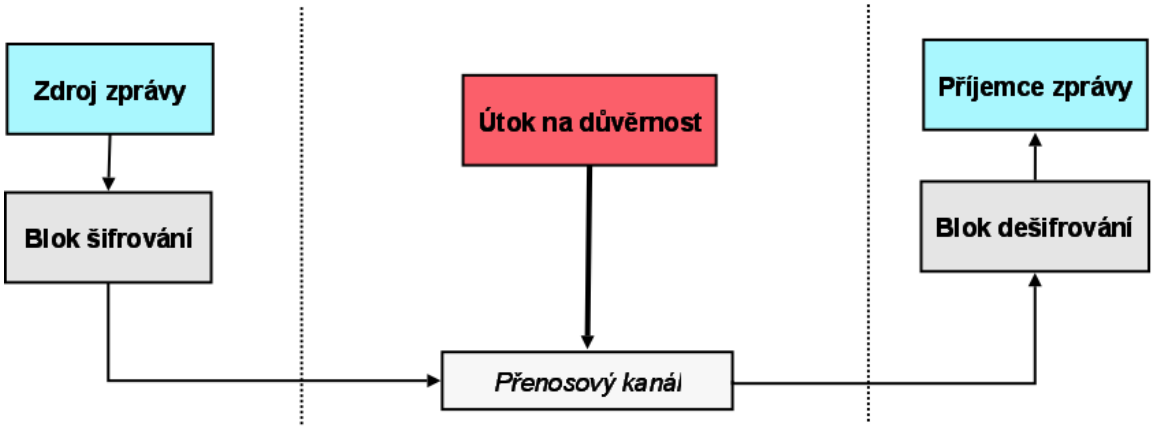
Útoky můžeme dělit na tyto druhy:

- Útok na autentičnost
- Útok na důvěrnost

Při útoku na autentičnost se útočník snaží podvrhnout data tím, že vstoupí mezi komunikující strany a snaží se jedné ze stran vnutit svoje data. Tomuto útoku se můžeme bránit například digitálním podpisem. Jedná se o rozsáhlou problematiku obsahující postupy zabezpečení pomocí vkládání hašů (haš je podobný kontrolnímu součtu, je vypočítán z dat a je jejich unikátním reprezentantem), certifikátů a veřejných klíčů [11, 13].

Tato práce blíže zkoumá praktické možnosti, kterými lze zajistit důvěrnost přenášených zpráv. Útok na důvěrnost značí, že se cizí osoba snaží získat data, které nejsou určena pro ni. K těmto datům mají mít přístup pouze autorizované osoby. Zabezpečení na úrovni veřejné datové sítě lze řešit pomocí šifrovacích metod.

Princip útoku na důvěrnost přenášené zprávy je nastíněn na obrázku 5.1. Útočník má k dispozici přenosové médium a je schopen získávat přenášenou komunikaci. Použité šifrování uživatelů se mu snaží zabránit v úspěšném provedení kryptoanalýzy a odvození použitého klíče, kterým by si zpřístupnil přenášená data [11].



Obrázek 5.1: Princip útoku na důvěrnost přenášené zprávy.

5.1 Základní druhy šifrování

Základní dělení šifrovacích mechanizmů:

a) Symetrické šifrování
b) Asymetrické šifrování

5.1.1 Symetrický systém

Jeho výhodou je rychlost použitých algoritmů. Je mnohonásobně rychlejší než asymetrický systém, z čehož vyplývá, že toto šifrování lze provádět v reálném čase. Z principů použitých šifrovacích kódů však plyne nižší odolnost proti kryptoanalýze ve srovnání s asymetrickými metodami. Symetrické šifrování rozlišujeme:

- Proudová šifra
- Bloková šifra

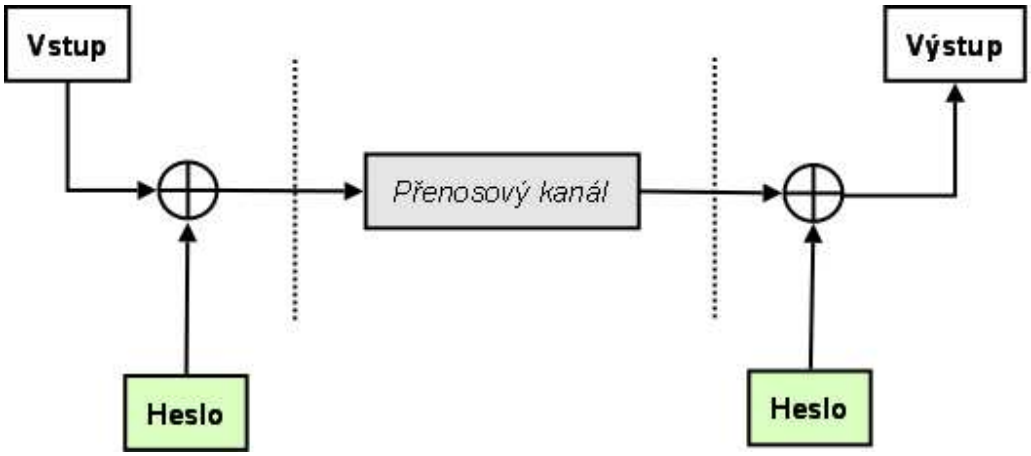
a) Proudová symetrická šifra

Pro nejjednodušší aplikace, kde není kladen důraz na bezpečnost, ale na rychlost a výpočetní nenáročnost šifrovacího algoritmu, je vhodná proudová šifra. Ta pracuje s daty jako s proudem bitů.

Do šifrovacího mechanismu vstupuje heslo, které tvoří posloupnost bitů a ty jsou jeden po druhém sečteny s příslušným bitem zprávy. Toto heslo je zpravidla získáváno generátorem pseudonáhodné posloupnosti se soustavou posuvných registrů a zpětných vazeb, do kterého je jako výchozí hodnota zadán tajný klíč. Následně je zpráva odesílána po přenosovém médiu.

Pro dešifrování platí obdobný mechanismus, kdy jsou jednotlivé bity kryptogramu sečteny s proudem bitů odpovídajícím vstupnímu heslu.

Z hlediska reálné využitelnosti jsou proudové šifry stále používané a to například v A5 zabezpečení technologie GSM nebo u WEP klíčů technologie 802.11.



Obrázek 5.2: Princip proudové šifry.

b) Bloková symetrická šifra

Bloková šifra je již považována za bezpečnou a to v závislosti na použitém šifrovacím algoritmu. Obecně platí, že bloková šifra pracuje s pevně stanoveným počtem bitů neboli blokem dat a neměnnými transformačními funkcemi. Délka jednotlivých bloků dat je zpravidla 64, 128, nebo 256 bitů.

Základem této šifry jsou blokové operace, které jednomu nebo více blokům bitům přiřazují jeden výstupní blok bitů. Používané druhy operací jsou [12]:

- Permutační
- Substituční
- Aritmetické

Permutace je libovolné přeuspořádání bitů. Permutace *n* prvků je skupina *n* prvků, které jsou uspořádány v jakémkoliv možném pořadí. Obecně jde o bijektivní zobrazení z množiny ***A*** na množinu ***B***. Permutace jsou buď s opakováním (libovolný prvek *x* z množiny ***A*** se může v cílové množině ***B*** vyskytnout vícekrát) nebo bez opakování (každý bit *x* z množiny ***A*** se může v množině ***B*** vyskytnout pouze jednou). Dále rozlišujeme permutaci prostou, kdy počet prvků množiny ***A*** je roven počtu prvků výsledné množiny ***B***, zúženou, kdy počet prvků množiny ***A*** je větší než počet prvků množiny ***B***, nebo rozšířenou, kdy počet prvků množiny ***A*** je menší než počet prvků množiny ***B***.

Častým typem permutace používaným v šifrovací praxi je rotace. Rotaci rozlišujeme pravou (označovanou ROR) nebo levou (označovanou ROL). Rotace zajišťuje posun o *n* prvků směrem daným typem rotace.

Substituce je zobrazení z množiny ***A*** na množinu ***B***. Jde o náhradu prvku za prvek jiný a tato záměna je definována substituční tabulkou, která obsahuje informace o tom, jakým prvkem z množiny ***B*** se bude který prvek z množiny ***A*** nahrazovat.

Aritmetické operace jsou operace, kdy vstupním blokům *X*, *Y* přiřazujeme výstupní blok *Z*. Velmi často jsou při šifrování využívány operace XOR a bitový součet [13].

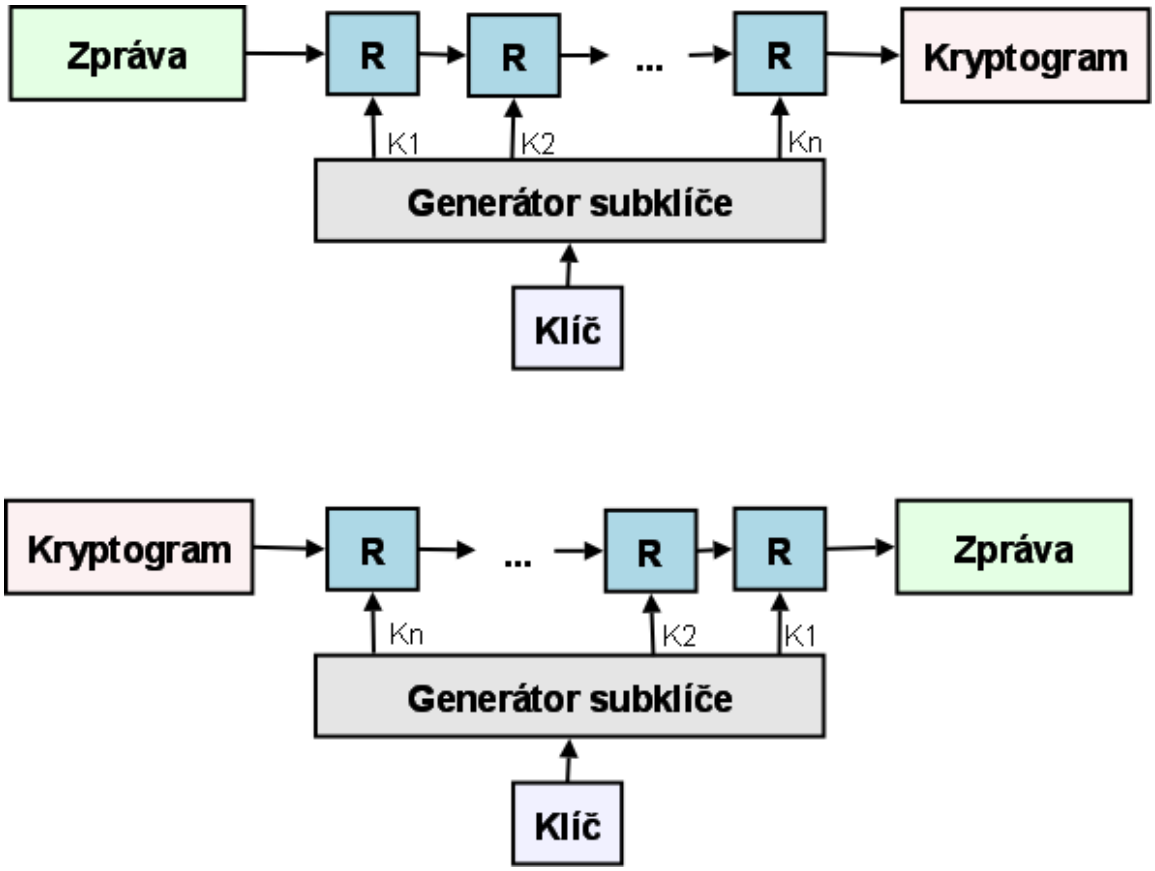
Mezi první komerčně velmi úspěšné blokové šifry patří šifra DES (*Data Encryption Standard*), vyvinutá firmou IBM a standardizovaná roku 1977. DES má blok dat o velikosti 64 bitů a velikost klíče 56 bitů. Tato šifra však již nesplňuje současné bezpečnostní nároky (oficiální prolomení v roce 1997). Nadstavbou této šifry je Triple DES. Principielně odpovídá trojnásobnému zřetězení původní šifry DES na každý blok dat. Podporuje délku klíče 56, 112 nebo 168 bitů.

V dnešní době je nejvíce komerčně využívanou blokovou šifrou AES šifra. Je charakteristická velmi dobrou odolností proti kryptoanalýze a v současnosti je většina komerčních aplikací využívajících blokových šifer postavena právě na této šifře (např. IPSec, SSL). Dosud není známo úplné prolomení této šifry.

Po pěti letech práce na výběru nejlepšího algoritmu pro standard AES byl nakonec vybrán Rijndael, systém symetrického blokového šifrování. AES byl dokončen a schválen roku 2001 americkým úřadem pro standardizaci. Délka bloku dat u šifry AES činí 128 bitů. Délka klíče může (dle zvoleného podtypu AES) být 128, 192 nebo 256 bitů.

Z dalších blokových šifer zmiňme alespoň Twofish, TripleFish, RC6, Serpent, Mars, Tea. Šifrou Tea se budeme blíže zabývat v praktické části této práce.

Princip blokové šifry je vyznačen na obrázku 5.3. Blok zprávy je šifrována pomocí kaskády samostatných elementárních blokových šifrátorů, tzv. rund. Každá runda přiřazuje svému vstupnímu bloku dat na základě svého subklíče, který je odvozen od celkového klíče zprávy, výstupní blok. Ten je přiveden na vstup následující rundy. Počet těchto elementárních šifrátorů je závislý na použitém typu šifrování [13].



Obrázek 5.3: Princip kódování a dekódování blokovou šifrou.

5.1.2 Asymetrický systém

Jeho výhodou je vyšší odolnost proti kryptoanalýze než u symetrického systému. Avšak je naopak pomalejší a neumožňuje kryptování a dekryptování v reálném čase. Tento systém je využíván k podepisování (zajištění autentičnosti) nebo šifrování (zajištění důvěrnosti) malých objemů dat.

5.2 Šifra XTEA

5.2.1 Historie vzniku šifry XTEA

Tato šifra vychází z původní šifry TEA (*Tiny Encryption Algorithm*) a je jejím vylepšeným následníkem. TEA je bloková šifra vynikající svou jednoduchostí a přitom dostatečnou kryptoanalytickou odolností. Byla vytvořena D. Wheelerem a R. Needhamem a byla poprvé prezentována roku 1994. Po nějaké době byly odhaleny slabiny této šifry a ty se staly podnětem pro vytvoření nové šifry XTEA, jenž z původní přímo vychází, avšak upravuje šifrovací aparát při zachování stejného stupně výpočetní náročnosti. Tato šifra byla prezentována roku 1997. Následujícího roku byla uveřejněna zatím poslední a nejodolnější verze této šifrovací rodiny – šifra XXTEA. Ta však doznala i větší složitosti a tím i úměrně vyšším nárokům na implementaci kódu a na výpočetní nároky hardwaru.

Velmi pozitivní vlastností šifer rodiny TEA je, že nepodléhají žádným patentním omezením [9].

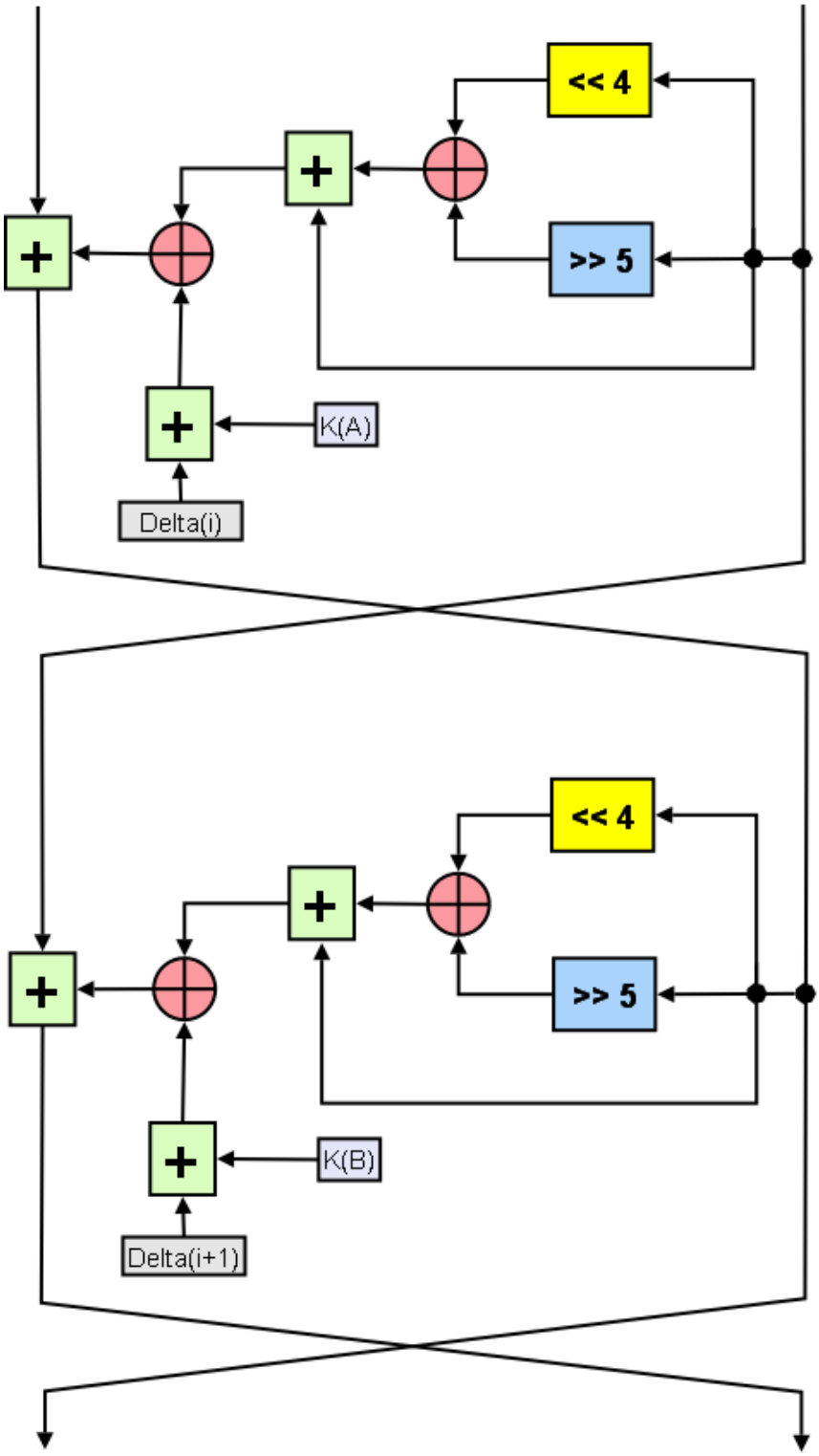
5.2.2 Rozbor šifry XTEA

Tato šifra pracuje s bloky dat o délce 64 bitů a využívá klíče délky 128 bitů. Její strukturu tvoří zřetězené Feistelovy rundy. Dvě tyto rundy tvoří jeden šifrovací cyklus. Šifra používá 32 kaskádně řazených cyklů (tj. 64 rund). Aby se zabránilo útokům založeným na symetrii Feistelových rund, jsou při získávání subklíčů pro jednotlivé cykly tyto klíče upraveny násobkem konstanty (označované jako *Magic Constant*). Tato konstanta je číselně rovna 2654435769 v desítkové soustavě nebo 9E3779B9 v soustavě šestnáctkové a je volena jako hodnota $2^{32}/\varphi$, kde φ je zlatý střed a je číselně roven $\varphi = 1+5^{1/2}/2$.

Na obrázku 5.4 je znázorněn princip šifrování touto šifrou. Do rundy šifry vstupuje blok dat jako dvě samostatné datové jednotky, každá o velikosti 32 bitů. Ke vstupu první části je přičten výsledek operací nad druhou částí bloku. Druhá část dat je přímo předána do další rundy, ale dále jsou nad druhou částí bloku prováděny aritmetické a permutační operace právě pro přičtení k části první. Tyto operace jsou následující:

Vstup je jednak bitově posunut doprava o čtyři pozice, jednak posunut doleva o pět pozic a výsledky jsou spolu násobeny funkcí XOR (e*xclusive OR*). Tato hodnota je bitově sečtena s původní vstupní hodnotou. Výsledná hodnota je XORována s patřičným subklíčem, ke kterému byla přičtena konstanta násobku zlatého řezu.

Pro následující rundu je výstup první části bloku předchozí rundy přiveden na vstup druhé části bloku a výstup druhé části bloku na vstup první části bloku.



Obrázek 5.4: Princip funkce jednoho cyklu XTEA šifry.

6. Konkrétní implementace interaktivity u DVB-T; vytvoření MHP aplikace

6.1 Definice požadavků na konkrétní MHP aplikaci

Vytvořená aplikace je spustitelná v prostředí DVB-T standardu na set-top boxu uživatele. Tato aplikace umožňuje získávání statistických údajů na libovolné téma a to formou ankety, přičemž obsah ankety je ovlivnitelný bez zásahu do zdrojového kódu MHP aplikace.

Data statistického průzkumu jsou uložena na vzdáleném serveru, který umožňuje i jejich libovolnou editaci. Tyto údaje se přenáší pomocí zpětného kanálu set-top boxu do uživatelem spuštěné MHP aplikace. Výsledky ankety jsou odeslány na server, vyhodnoceny a uživatel je informován přehlednou formou o aktuálním stavu ankety. Jako nejefektivnější způsob zobracení těchto přijatých výsledků se za daných podmínek jeví graf, proto je využit i v této aplikaci.

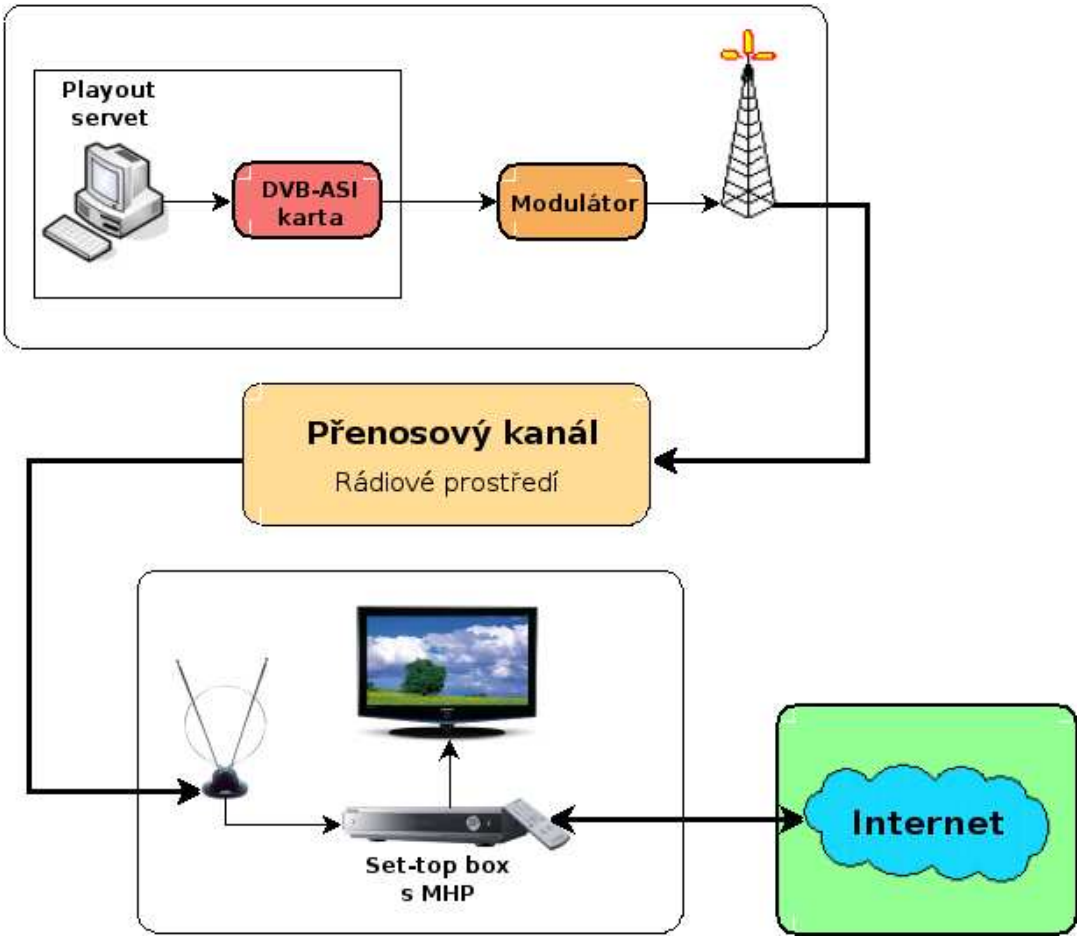
Při komunikaci mezi serverem a uživatelovým set-top boxem je využíváno šifrovaného spojení. Dále je pro jednoznačné ověření uživatelů navrhnut systém správy a identifikace uživatele na základě unikátního číselného kódu.

Server si vede databázi informací o jednotlivých přístupech pro potřeby dalšího vyhodnocení. Je ukládána informace o přístupu obsahující jednoznačný i dodatkový popis uživatele, čas přístupu i jeho volbu, a zda byl jeho hlas akceptován.

Celá aplikace je prezentována formou uživatelsky příjemného prostředí s propracovanou grafikou, s podporou vkládání bitmapových obrázků a dynamického vykreslování výsledků ankety.

6.2 Hardwarové požadavky při tvorbě MHP aplikací

Praktická část této práce se zabývá vytvořením a zprovozněním funkční MHP aplikace. Ta má být využitelná v reálném prostředí digitálního televizního řetězce, kde na vysílací straně je vložena společně s video složkou jednotlivých televizních pořadů a dalších vysílaných MHP aplikací do DVB-T datovém streamu a na přijímací straně je možné ji přijmout a spustit na reálném set-top boxu.



Obrázek 6.1: DVB-T pracoviště podporující MHP aplikace.

Samotná realizace laboratorního prostředí schopného pracovat s MHP aplikacemi je na obrázku 6.1. Na tomto DVB-T řetězci je možné testovat MHP aplikaci téměř v reálných podmínkách. Vysílací strana se skládá z PC s příslušným výstupním rozhraním, tím je DVB-ASI karta. V počítači je nainstalována aplikace *Playout server*, což je program, který ze zvoleného video signálu a požadovaných aplikací vytvoří jeden datový tok (tzv. *ASI* výstup) a ten odesílá na své výstupní rozhraní. Dále je datový tok veden na modulátor, kde je kódován a modulován na požadovanou nosnou, zesílen a pak již pomocí vysílací antény odeslán do rádiového prostředí.

Na přijímací straně je signál nejdříve zachycen vhodnou anténou a odeslán do set-top boxu schopného pracovat s MHP. Je třeba, aby set-top box podporoval standard MHP 1.1, a to kvůli použitému druhu komunikace využívající služeb aplikačních protokolů. Data jsou na tomto set-top boxu zpracována a grafická reprezentace signálu je odesílána na televizní obrazovku, kde je vykresleno uživatelské rozhraní aplikace. Set-top box také spravuje přijímané MHP aplikace a buď je automaticky spustí, nebo pomocí dálkového ovladače je může spustit sám uživatel. Set-top box je dále napojen do sítě internet a to pomocí standardní ethernetové přípojky. To umožní aplikaci být plně interaktivní a tudíž jak získávat data z internetu, tak je i na něj odesílat.

6.3 Softwarové požadavky při tvorbě MHP aplikace

Pro vytváření MHP aplikace je vhodné použít kvalitní prostředí pro vývoj programů na základě programovacího jazyku Java. Pro potřeby této MHP aplikace byl využit vývojový a odlaďovací nástroj JCreator 4.00.

K podpoře a umožnění správného překladu zdrojového kódu a zobrazování grafického režimu je využíván balíček Java JDK verze 1.4.2. Je nutné pro chod MHP aplikace zvolit vhodný druh Java JDK balíčku, protože některé novější verze neumožňují správnou kompilaci zdrojového kódu.

Pro potřeby této MHP aplikace jsou využity některé další knihovny, rozšiřující možnosti a dovolující základní chod aplikace. Těmto knihovnám se blíže věnuje kapitola 6.4.

6.3.1 Java JDK

Java JDK (*Java Development Kit*) je soubor nástrojů pro vývoj aplikací pro platformu Java. Je licencován pod GNU General Public Licence a patří pod opensource projekt OpenJDK [10].

Jeho součástí jsou:

- JRE (*Java Runtime Environment*) – spouštění nástrojů aplikací; obsahuje virtuální stroj a sadu základních knihoven Java Core API.
- Překladač *javac* ze zdrojového kódu java do kódu strojového.
- *Debugger* – odlaďovací nástroj.
- Nástroj pro vytváření *jar* archivů.
- Další pomocné nástroje.

6.4 Rozšiřující knihovny konkrétní MHP aplikace

6.4.1 Rodina knihoven java.awt.*

Poskytuje základní prvky uživatelského rozhraní včetně využití práce s grafikou.

java.awt.event.KeyEvent – Poskytuje nízkoúrovňové rozhraní mezi klávesnicí a aplikací.

java.awt.event.KeyListener – Umožňuje vytvoření objektu *addKeyListener*, který obsluhuje události stisku klávesy, uvolnění, nebo stisk společně s uvolněním.

java.awt.Color – Zajišťuje definici a správu barev. Jednak obsahuje některé předdefinované barvy, umožňuje ale i definici vlastní barvy. Podporuje barevné palety RGB, sRGB a HSB.

java.awt.Component – Knihovna umožňuje vytvoření vlastního grafického objektu, jeho vykreslení a interakci s okolím.

java.awt.Font – Knihovna rozšiřuje možnosti reprezentace textu.

java.awt.Graphics – Knihovna umožňuje vykreslovat základní i rozšířené geometrické tvary.

java.awt.Toolkit, *java.awt.MediaTracker*, *java.awt.Image* – Knihovny pro využití obrázků.

6.4.2 Knihovny pro síťovou komunikaci

*java.net.** – Knihovna pro definování síťové komunikace.

*java.io.** – Knihovna umožňující přijímání a odesílání datového streamu.

6.4.3 Knihovny pro práci s Xletem

javax.tv.xlet.Xlet – Knihovna poskytuje aplikačnímu manažeru možnost provádět základní operace životního cyklu Xletu (create, initialize, start, pause, destroy).

javax.tv.xlet.XletContext – Knihovna zpřístupňuje Xletu informace o tom, v jaké stavu se aplikace právě nachází.

javax.tv.xlet.XletStateChangeException – Knihovna pro obsloužení vyjímek vzniklých při běhu Xletu.

6.4.4 Knihovny rodiny HAVi

Knihovna základních i rozšiřujících funkcí pro konkrétní implementaci grafiky na zařízeních typu set-top box.

org.havi.ui.HScene – Knihovna zpřístupňuje přímé vykreslování na obrazovku a poskytuje spojení s nastavením manažera oken konkrétního zařízení.

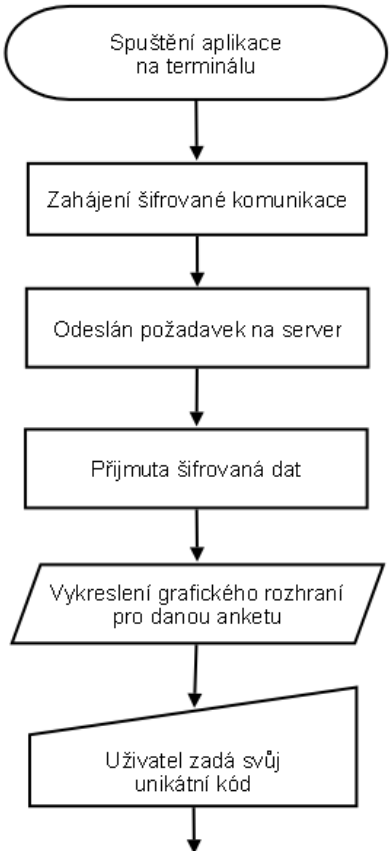
org.havi.ui.HSceneTemplate – Knihovna popisuje jednotlivé parametry zobrazované scény.

org.havi.ui.HsceneFactory – Knihovna poskytuje mechanismy pro nastavení parametrů zobrazení, které budou co nejblíže požadovaným hodnotám a budou vycházet z potřeb daného zobrazovacího hardwaru.

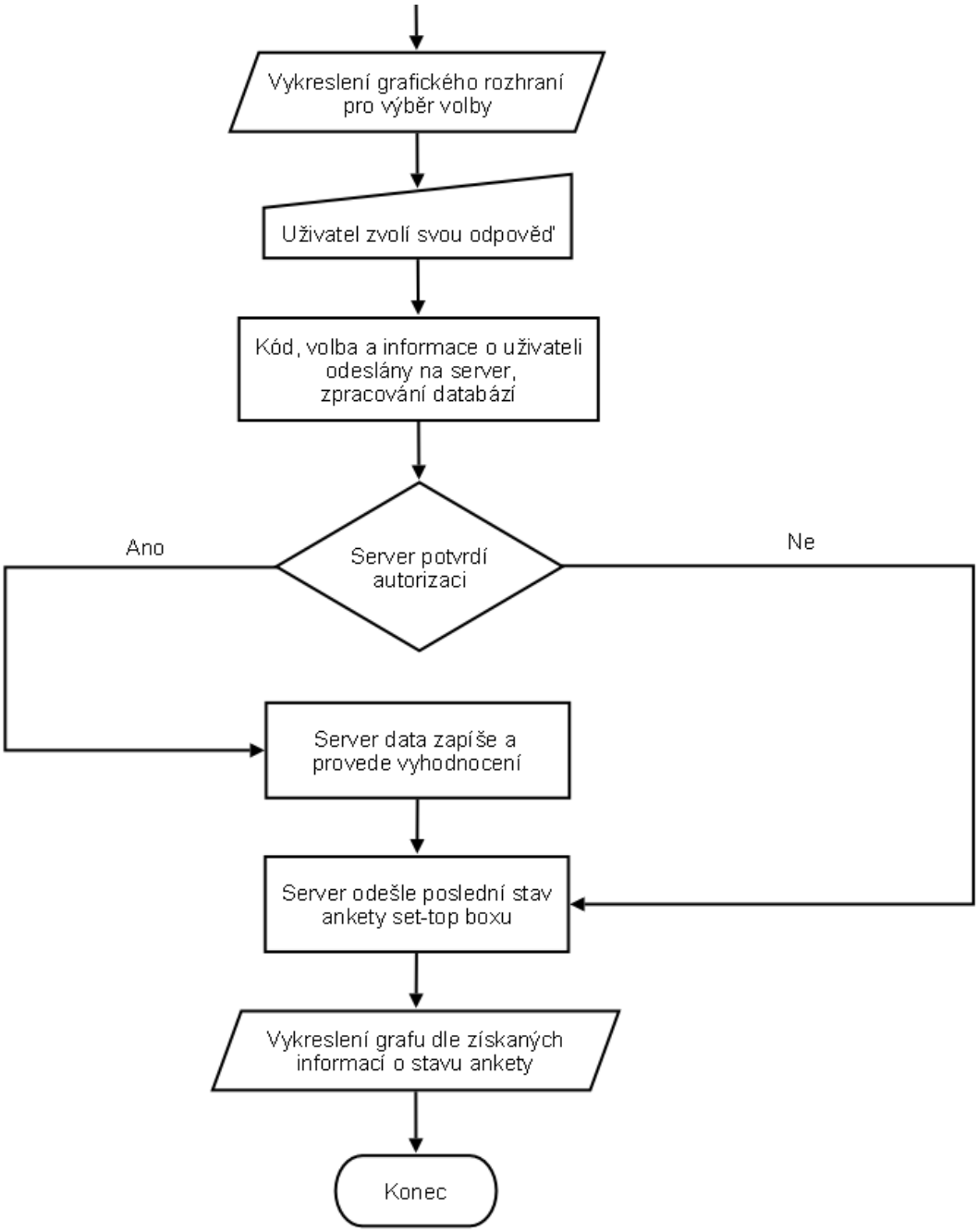
6.5 Vývojový diagram aplikace

Vývojový diagram aplikace symbolicky znázorňuje postup při načítání, vyhodnocování a zpracování dat v rámci celého algoritmu. Z tohoto vývojového diagramu je zřejmé, že je snaha o co největší přesunutí výpočetní náročnosti na server a minimalizování nutných systémových prostředků daného set-top boxu. Set-top box pouze přijímá, odesílá a vykresluje data; o vyhodnocení informací se stará strana serveru.

Co se týče jednotlivých bloků diagramu, budou blíže popsány z hlediska konkrétní implementace v následujících kapitolách a to jak ze strany serveru, tak set-top boxu.



Obrázek 6.2: Vývojový diagram MHP aplikace, první část.



Obrázek 6.3: Vývojový diagram MHP aplikace, druhá část.

6.6 Spuštění aplikace, inicializace scény, životní cyklus

Jak již bylo popsáno v teoretické části práce, MHP aplikace podléhá životnímu cyklu. Proto při spuštění aplikace na set-top boxu je nejdříve správcem aplikací vytvoří instance třídy a to zavoláním jejího konstruktoru. V běžných Xletech se v konstruktoru neprovádějí žádné inicializace či nastavování proměnných a tyto činnosti jsou obsaženy v metodě *initXlet()*, ale lze i této možnosti využít.

Po zavolání konstruktoru následuje volání metody *initXlet()*. Ta obsahuje základní volání metod, vytváření instancí tříd a aplikace hodnot proměnných pro připravení Xlet aplikace pro spuštění.

V této konkrétní MHP aplikaci se provede vytvoření nové scény typu HScene a vybrání nejvhodnějších parametrů této scény pomocí metody *getBestScene()*. Následně je volána třída *myClass* a je vytvořena její instance se zvolenými parametry velikosti dané scény. Ty jsou omezeny maximálními zobrazovacími schopnostmi hardwarového vybavení. Tato instance je přidána do vytvořené scény, avšak ještě předtím je zavolána metoda *nacti_data_do_struktury()* z třídy *myClass*, která zajistí získání dat pro vykreslení aktuální ankety z předem definovaného serveru.

Metoda *destroyXlet()* uvolňuje systémové prostředky alokované pro zobrazovanou scénu.

Důležitou metodou je veřejná metoda *keyPressed (KeyEvent)*, jenž zpřístupňuje vstupní uživatelské rozhraní, v tomto případě dálkové ovládání set-top boxu, a předává informace o případném zmáčknutém tlačítku. Tato metoda je definována v knihovně *java.awt.event.KeyListener*, vstupní proměnná typu *keyEvent* pak v knihovně *java.awt.event.KeyEvent*.

6.7 Šifrování komunikace

Dle požadavků na tuto konkrétní aplikaci bylo nutné implementovat vhodný druh ochrany přenášených dat použitím adekvátní šifrovací metody.

Po analýze problematiky a při zahrnutí požadavku hardwaru na co nejméně náročnou šifrovací metodu, avšak poskytující dostatečný stupeň zabezpečení, byla vybrána jako vhodná šifrovací metoda XTEA, z rodiny šifry TEA. Teoretický rozbor zpracovává kapitola 5.2.

Kódová implementace šifry XTEA není v současné době pro potřeby této aplikace v nějaké vhodné formě dostupná, na rozdíl například od knihoven pro práci s RSA šifrou, AES, apod., a proto musela být zcela kompletně pojata a šifrovací metodika přepsána do zdrojového kódu. Pro potřeby Xletu byla dle teoretických principů šifry vytvořena třída *myXTEA*, jenž zapouzdřuje práci s touto šifrou.

6.7.1 Minimum pro práci v jazyce Java a PHP z hlediska implementace šifrovacích procedur

Abychom byli schopni pracovat na spolupráci jazyků Java a PHP, je nutná znalost odlišností v definovaných datových typech a teorie používaných operátorů.

a) Datové typy

V jazyce Java se proměnné deklarují zvolením datového typu a názvem proměnné. Možné datové typy a jejich velikost udává tabulka 6.1.

Tab. 6.1: Datové typy jazyka Java.

typ	popis	velikost	min. hodnota	max. hodnota
byte	celé číslo	8 bitů	-128	+127
short	celé číslo	16 bitů	-32768	+32767
int	celé číslo	32 bitů	-2147483648	+2147483647
long	celé číslo	64 bitů	-9223372036854775808	+9223372036854775807
float	reálné číslo	32 bitů	-3.40282e+38	+3.40282e+38
double	reálné číslo	64 bitů	-1.79769e+308	+1.79769e+308
char	znak UNICODE	16 bitů	/u0000	/uFFFF
boolean	logická hodnota	1 bit	-	-

V jazyce PHP je přístup k deklaraci proměnné trochu odlišný. Explicitní zvolení datového typu proměnné není nutné. Typ proměnné se volí podle typu ukládané hodnoty do této proměnné.

Omezení hodnoty ukládané do proměnné je dáno použitou platformou, nad kterou jazyk PHP běží, ale obecně jsou proměnné téměř bezlimitní pro běžné použití. Pro reálné číslo v jazyku PHP se udává přesnost na 14 desetinných míst.

Rozdílnost možných rozsahů používaných datových typů mezi jazyky Java a PHP je nutné drát v úvahu při navrhování aplikace pro vzájemnou kooperaci obou jazyků.

b) Operátory bitového posunu

Operátor >> reprezentuje bitový posun vpravo s rozšířením znaménka. Deklarace složena z proměnné před operátorem, která udává hodnotu, na kterou má být aplikován bitový posun a proměnné za operátorem udávající o kolik pozic má být posun proveden. Bity, které jsou přesunuty doprava, jsou vypuštěny. Operátor zachovává znaménko, takže bity na levé straně budou vyplněny nulami, jestliže nejdůležitější bit (MSB) je roven nule a vyplněny hodnotou jedna, jestliže je nejdůležitější bit roven jedné. Přesunutí hodnoty doprava o jednu pozici se rovná dělení dvěma s případným vypuštěním zbytku.

Příklad:

65535 >> 8 = 255

0000000000000000111111111111111 binárně (65535 desítkově)

>> 8 =

00000000000000000000000011111111 binárně (255 desítkově)

-8 >> 1 = -4

11111111111111111111111111111000 binárně (-8 desítkově)

>> 8 =

11111111111111111111111111111100 binárně (-4 desítkově)

Operátor << reprezentuje bitový posun vlevo s rozšířením znaménka. Deklarace složena z proměnné před operátorem, která udává hodnotu, na kterou má být aplikován bitový posun a proměnné za operátorem udávající o kolik pozic má být posun proveden. Bity, které jsou přesunuty doleva, mimo levý konec jsou vypuštěny a bity vpravo, které jsou vyprázdněny přesunem na jinou pozici, jsou vyplněny hodnotou nula. Hodnota posunutí doleva o jednu pozici je ekvivalentem vynásobení dvěma.

Příklad:

1 << 10 = 1024

00000000001 binárně (1024 desítkově)

<< 10 =

01111111111111111111111111111111 binárně (255 desítkově)

Operátor >>> reprezentuje stejnou permutaci bitů jako bitový operátor **>>** s výjimkou toho, že nezachovává znaménko původního výrazu, protože bity vlevo se vždy vyplňují nulami [10].

Příklad:

-1 >>> 1 = 2147483647

11111111111111111111111111111111 binární (-1 desítkově)

<< 10 =

10000000000 binární (255 desítkově)

6.7.2 Aplikace principů šifry XTEA na vývojové prostředí Java

Potřeby šifrování na straně uživatele, tj. MHP aplikace, jsou obsluhovány pomocí třídy *myXTEA*. Před prvním použitím této šifrovací třídy je nutné vytvořit instanci třídy. Poté je metodou *public void inicializuj (String)* této instanci předán klíč o délce 128 bitů a to ve formátu String (to znamená 16 znaků). Tento klíč je rozložen na čtyři čísla typu integer a velikosti 32 bitů. K tomu slouží metoda *private int bytes2Int(byte[], int t)*, kde prvním vstupním parametrem jsou data a druhým offset, tj. posunutí od počátku hodnoty vyjádřené v bajtech. Nyní je instance připravena šifrovat či dešifrovat data. Tato data vstupují taktéž ve formátu String.

Šifrovací funkcí, která je vnějším rozhraním pro šifrování je metoda *public String sifruj(String)*. Do ní uživatel třídy vloží svá data a výstupem mu je datový řetězec šifrovaných dat. V této metodě jsou příchozí data rozdělena na 64 bitové bloky a v případě, že poslední blok neobsahuje všech 64 bitů, je zavolána metoda *private String uprav_delku (String)*. Tato metoda se postará o případné přidání bílých znaků, tzv paddingu. Následně jsou jednotlivá data odesílána metodě *private String sifruj_blok(String)*, ta jednotlivé bloky zašifruje a vrátí metodě *public String sifruj(String)*, jenž bloky poskládá a vrátí uživateli.

Metoda *private String sifruj_blok(String)* si nejprve pomocí již popsané metody *private int bytes2Int(byte[], int t)*, rozloží blok dat na dvě čísla typu integer. Následuje hlavní cyklus šifrování schematicky popsaný na obrázku 5.4. Zde je vhodné zdůraznit význam operátorů <<, >>, >>>. Zatímco operátory <<, >> reprezentují bitový posun vlevo a vpravo s rozšířením znaménka, operátor >>> představuje bitový posun vpravo s rozšířením nuly (*Unsigned Right Shift*). Blíže o tomto problému pojednává kapitola 6.7.3. Toto má konkrétní dopad na výslednou hodnotu těchto permutačních operací. Následuje převod získaných hodnot do podoby datového řetězce a to pomocí metody *private void Int2bytes(int, int[], int)*. Vstupem je převáděné číslo, datový řetězec reprezentovaný polem integerů a již zmiňovaný offset. Intuitivně by šlo předpokládat, že na definici číselné reprezentace pole znaků bude použit celočíselný typ byte. Pole integerů je použito proto, že jazyk Java definuje celočíselný typ pro hodnoty od – 128 do 127, přičemž pro převod na char požadujeme rozmezí hodnot od 0 až po 256.

Opačný proces, kterým je dešifrování se řídí obdobným postupem, ovšem inverzně. Na konci dešifrování jsou odstraněny případné bílé znaky funkcí *trim()*.

Z pomocných metod této třídy je nutné zmínit ještě metodu *private String pridej_ch13(String)*, která byla přidána po testování přenosu dat mezi serverem. Zde po určité době docházelo k přenosu chybných zpráv a následnému rozpadu synchronizace dat vedoucímu ke ztrátě celé následné zprávy. To bylo způsobeno formou přenosu, kdy aplikace načítá ze serveru data po řádcích a v případě, že se na řádku objevil znak s ordinální hodnotou 13 (tzv. *Carriage Return*), byl již zbytek řádku považován za řádek následující. Proto je tento znak na straně serveru odstraněn. Ale protože by toto odstranění vedlo k milnému dešifrování dat, je na místě jeho výskytu vložen libovolný zvolený znak, v tomto případě znak s ordinální hodnotou 65, tj. „A“. Kdekoliv byl v původních datech znak „A“, dojde k jeho zdvojení. Tím je poté na straně uživatele možné výskyt dvou znaků „A“ hned po sobě nahradit jedním znakem „A“ a osamocený výskyt znaku „A“ rozpoznat jako *Carriage Return* a tímto znakem nahradit. O toto se na straně uživatele stará metoda *private String pridej_ch13(String)*.

6.7.3 Aplikace principů šifry XTEA na vývojové prostředí PHP

Na straně serveru byla vytvořena knihovna poskytující službu šifrování XTEA šifrou. Tato šifrovací knihovna funguje na obdobném principu jako již popsaná alternativa v jazyku Java, proto se budeme zabývat jen odlišnostmi v implementaci do jazyka PHP.

Prvním odlišností je druh a způsob používání tohoto šifrovacího aparátu. V jazyce Java byla vytvořena třída, kde se až při použití vytvořila instance třídy, rezervovala paměť a inicializační metodou nastavila hodnota šifrovacího klíče. Pro potřeby šifrování v jazyce PHP byl tento šifrovací aparát pojat jako samostatný soubor obsahující realizaci jednotlivých funkcí. Následně je v jakémkoli kódu, kde je zapotřebí šifrování, nutné zajistit, aby pomocí příkazu *include* byl importován tento soubor se zdrojovým kódem šifrovacích funkcí a pak už stačí volat jednotlivé funkce pro jednotlivé procesy. Odpadá tedy vytváření instancí a prvotní inicializace. Pro zahájení šifrování stačí tedy použít funkci

function sifruj($data, $klic), kde je vidět, že jako vstupní parametr vstupuje jak datový řetězec, tak řetězec klíče.

Klíč je rozložen na použitelný formát číselné reprezentace při každém zavolání funkce šifrování, avšak toto nemá žádný negativní dopad na rychlost aplikace, protože kód je aplikován na straně serveru, kde je dostupný daleko vyšší výpočetní výkon, než na straně set-top boxu.

Zásadní rozdíl doznal hlavní výpočetní cyklus XTEA šifry vycházející z principu znázorněného na obrázku 5.4, a to z důvodu, že programovací jazyk je od jazyku Java rozdílný v deklarovaných číselných typech. Dále se tyto jazyky liší některými funkcemi. Jazyk PHP neposkytuje operátor >>>, bitový posun vpravo s rozšířením nuly, a proto tento operátor musel být zastoupen vytvořením nové funkce. Tato funkce se jmenuje *function urshift($cislo, $o_kolik)* a vrací právě hodnotu jako operátor >>> v jazyce Java. Kromě funkce pro náhradu tohoto operátoru bylo nutné vytvořit funkci pro bitový součet dvou čísel. Jedná se o funkci *function soucet_b($i1, $i2)*.

6.7.4 Ověření šifrování na síťové vrstvě

Pro demonstraci šifrování na přenosovém médiu byl použit analyzátor síťového provozu WireShark 0.99.8. Díky němu byla získána následná data.

Síťový provoz aplikace byl zkoumán ve dvou krocích – bez použití šifrování a s šifrováním. Výsledek je zřejmý – šifrování znemožňuje přímou interpretaci dat ze strany útočníka aniž by prolomil použitou šifru.

a) Nešifrovaný přenos (protokol HTTP)

```
Time        Source           Destination      Protocol    Info
4. 388720   88.86.113.138    192.168.1.12     HTTP        HTTP 1.1 OK

Frame 60 (284 bytes on wire, 284 bytes captured)
Ethernet II, Src: SmcNetwo_43:e5:ea (00:13:f7:43:e5:ea), Dst:
CompalEl_5c:ee:03 (00:0f:
b0:5c:ee:03)
Internet Protocol, Src: 88.86.113.138 (88.86.113.138), Dst: 192.168.1.12
(192.168.1.12)
Transmission Control Protocol, Src Port: http (80), Dst Port: pciarray
(1552), Seq: 126
1, Ack: 159, Len: 230
[Reassembled TCP Segments (1490 bytes): #59(1260), #60(230)]
Hypertext Transfer Protocol
```

Reassembled TCP (1490 bytes):

```
0120 0a 33 0a 31 32 0a 31 33 0a 31 0a 73 74 69 6e 6f .3.12.13.1.stino
0130 76 61 6e 79 0a 62 69 6c 61 0a 33 36 0a 31 34 30 vany.bila.36.140
0140 0a 33 36 0a 50 f8 65 64 76 6f 6c 65 62 6e ed 20 .36.P.edvolebn.
0150 61 6e 6b 65 74 61 0a 74 65 78 74 0a 62 69 6c 61 anketa.text.bila
0160 0a 31 38 0a 34 33 0a 31 30 30 0a 56 20 20 72 e1 .18.43.100.V r.
0170 6d 63 69 20 20 76 fd 7a 6b 75 6d 75 20 20 70 72 mci v.zkumu pr
0180 65 66 65 6e 63 ed 0a 74 65 78 74 0a 62 69 6c 61 efenc..text.bila
0190 0a 31 38 0a 32 30 0a 31 32 35 0a 76 6f 6c 69 e8 .18.20.125.voli.
01a0 f9 20 20 56 e1 6d 20 20 70 f8 65 64 6b 6c e1 64 . V.m p.edkl.d
01b0 e1 6d 65 20 20 74 75 74 6f 0a 74 65 78 74 0a 62 .me tuto.text.b
01c0 69 6c 61 0a 31 38 0a 32 30 0a 31 35 30 0a 61 6e ila.18.20.150.an
01d0 6b 65 74 75 2c 20 64 ed 6b 79 20 6e ed 9e 20 62 ketu, d.ky n.. b
01e0 75 64 65 20 6d 6f 9e 6e 6f 20 6a 69 9e 0a 74 65 ude mo.no ji..te
01f0 78 74 0a 62 69 6c 61 0a 31 38 0a 32 30 0a 31 37 xt.bila.18.20.17
0200 35 0a 70 f8 65 64 20 20 f8 e1 64 6e fd 6d 69 20 5.p.ed ..dn.mi
0210 20 76 6f 6c 62 61 6d 69 20 20 6f 64 68 61 64 2d volbami odhad-
```

Line-based text data: text/plain

```
3\n
12\n
13\n
1\n
stinovany\n
bila\n
36\n
140\n
36\n
P\370edvolebn\355 anketa\n
text\n
bila\n
18\n
43\n
100\n
V r\341mci v\375zkumu prefenc\355\n
text\n
bila\n
18\n
20\n
125\n
voli\350\371 V\341m p\370edkl\341d\341me tuto\n
text\n
bila\n
18\n
20\n
150\n
anketu, d\355ky n\355\236 bude mo\236no ji\236\n
text\n
bila\n
18\n
20\n
175\n
p\370ed \370\341dn\375mi volbami odhad-\n
```

b) Šifrovaný přenos (protokol HTTP)

```
Time        Source            Destination       Protocol    Info
4.501944    88.86.113.138     192.168.1.12      HTTP        HTTP 1.1 OK

Frame 12 (185 bytes on wire, 185 bytes captured)
Ethernet II, Src: SmcNetwo_43:e5:ea (00:13:f7:43:e5:ea), Dst:
CompalEl_5c:ee:03 (00:0f:
b0:5c:ee:03)
Internet Protocol, Src: 88.86.113.138 (88.86.113.138), Dst: 192.168.1.12
(192.168.1.12)
Transmission Control Protocol, Src Port: http (80), Dst Port:
serialgateway (1243), Seq
: 3781, Ack: 173, Len: 131
[Reassembled TCP Segments (3911 bytes): #8(1260), #9(1260), #11(1260),
#12(131)]
Hypertext Transfer Protocol

Reassembled TCP (3911 bytes):
00a0 0a 65 39 61 0d 0a 0d 0a 78 68 50 c3 bc c3 b5 c2 .e9a....xhP.....
00b0 93 56 77 0a c3 a6 c2 ad c2 99 6b 3c 30 c2 a2 c2 .Vw.......k<0...
00c0 87 0a 4b c2 a4 61 71 c2 94 c2 a3 05 00 0a 2b c2 ..K..aq.......+.
00d0 b9 c2 9d 06 c2 a0 67 74 6d 0a 1b c2 8e c3 8e c3 ......gtm.......
00e0 8a c2 93 6c c2 a1 c2 bc c3 bf c3 b8 11 c2 ba c2 ...l............
00f0 9b c2 90 03 1a 0a c2 ac c3 bf 4c 62 54 c3 82 c2 ..........LbT...
0100 9b c3 9e 0a c2 9a c3 84 70 03 4f 34 c3 9e 45 0a ........p.O4..E.
0110 c3 a8 c3 bd c2 b6 c2 bb c3 ab c2 ab 40 c2 89 0a ............@...
0120 43 25 c2 b9 75 1f c2 b1 22 c2 af 0a c2 9d c3 91 C%..u..."........
0130 28 c2 a1 60 69 36 24 6e 71 0c 05 c3 b4 43 24 c3 (..‘i6$nq....C$.
0140 87 c3 8c c2 b9 63 65 c3 bf 54 c3 ba 1b 0a 15 2b .....ce..T.....+
0150 c3 a6 67 c2 99 c2 8b 56 51 0a c2 93 7a c3 8b c3 ..g....VQ...z...
0160 a5 c2 bd c2 b3 22 c3 81 0a 42 c2 90 1e c3 ad 64 ....."...B.....d
0170 c3 8f c2 96 2b 0a 5a 14 15 48 c2 87 c2 bb 60 21 ....+.Z..H....‘!
0180 0a c2 98 c3 ae 43 09 c3 92 c3 a2 58 5b 0a c3 96 .....C.....X[...


Line-based text data: text/html
\r\n
xhP\303\274\303\265\302\223Vw\n
\303\246\302\255\302\231k<0\302\242\302\207\n
K\302\244aq\302\224\302\243\005\000\n
+\302\271\302\235\006\302\240gtm\n
\033\302\216\303\216\303\212\302\223l\302\241\302\274\303\277\303\270\021
\302\272\302\2
33\302\220\003\032\n
\302\254\303\277LbT\303\202\302\233\303\236\n
\302\232\303\204p\003O4\303\236E\n
\303\250\303\275\302\266\302\273\303\253\302\253@\302\211\n
C%\302\271u\037\302\261"\302\257\n
\302\235\303\221(\302\241‘i6$nq\f\005\303\264C$\303\207\303\214\302\271ce
\303\277T\303\
272\033\n
\025+\303\246g\302\231\302\213VQ\n
\302\223z\303\213\303\245\302\275\302\263"\303\201\n
B\302\220\036\303\255d\303\217\302\226+\n
Z\024\025H\302\207\302\273‘!\n
```

6.8 Odeslání žádosti (*Request*) na server, odpověď (*Response*)

6.8.1 Implementace žádosti a odpovědi v jazyku Java

Komunikace s PHP serverem funguje na principu *Request – Response*. Ke generování výzvy je u uživatele volána metoda *public void nacti_data_do_struktury()*. Jako komunikační kanál je využíván protokol aplikační vrstvy HTTP. Ten je zpřístupněn instancí třídy URL z knihovny *java.net.URL*. Instance třídy *java.io.BufferRead* zpřístupní získání odpovědi od HTTP protokolu. Čtení probíhá se znakovým kódováním UTF-8. Data jsou postupně načítána, dešifrována a ukládána do předpřipravené struktury. Tato struktura je tvořena třídou *myRadek*, která uchovává informace o jednom řádku textu ankety. Jedná se o tyto informace:

- Typ zobrazovaného textu
- Požadovaná barva textu
- Požadovaná velikost
- Umístění, souřadnice x
- Umístění, souřadnice y
- Požadovaný text

Pro každý načtený řádek je dynamicky alokována paměť vytvořením dvourozměrného pole o proměnné velikosti (dle požadavků), kde položky pole tvoří instance třídy *myRadek*. První dimenze pole reprezentuje žádané umístění řádku v rámci celé ankety, tj. pořadové číslo strany. Druhá dimenze pole definuje pořadové číslo řádku v rámci dané strany.

Pole je voleno jako dynamicky alokované z důvodu zamezení zbytečného rezervování systémových prostředků.

Následně jsou pomocnou metodou *private void nacti_data_pro_sipku()* předána data o možnostech voleb, které bude uživatel v rámci ankety vybírat. Řádky, které tvoří volbu v rámci ankety, jsou načteny do obdobné struktury jako *myRadek*, v tomto případě s názvem *myVolba*. Tato struktura pro potřeby pozdějšího vyhodnocení ankety navíc obsahuje počet hlasů, které tato volba od uživatelů získala a pro snížení výpočetní náročnosti aplikace i počet procent hlasů dané volby.

Dále jsou do paměti načteny bitmapy grafického rozhraní ankety.

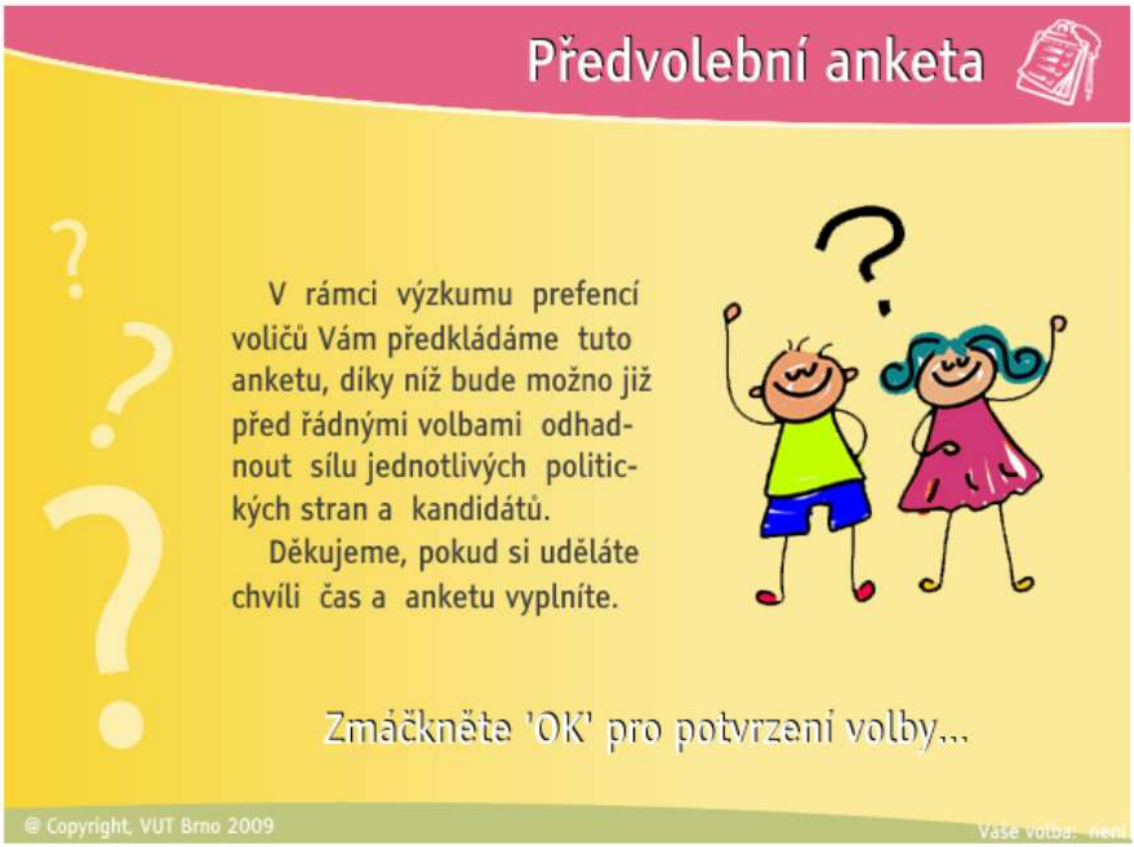
6.8.2 Implementace žádosti a odpovědi v jazyku PHP

Skript vyhodnocující žádost o hlavní data ankety je obsažen v souboru *hlavni_data_ankety.php*. Tento skript obsahuje přidání souboru pro podporu šifrování *myXtea.php*. Dále jsou definovány přístupové parametry k SQL databázi a je s ní navázána komunikace. Nyní skript na základě informací získávaných z SQL databáze předává HTTP protokolu jednotlivá již zašifrovaná data, která obsahují počet stran ankety, počet řádků na jednotlivých stranách a data jednotlivých řádků ankety. Data jsou prezentována v UTF-8 kódování.

6.9 Grafická prezentace vstupních dat ankety

Data, která MHP aplikace obdržela od PHP serveru, je nyní nutné prezentovat. K tomuto účelu slouží veřejná metoda *public void paint(Graphics)*, která je volána při žádosti manažera aplikací o vykreslení aplikace.

Jako první se v rámci této ankety vykreslí zvolené pozadí ve formě bitmapy. Následuje rozhodnutí, na které straně ankety se uživatel právě nachází. Anketa je navržena jako třístránková, kdy na první straně jsou prezentovány základní informace, na další straně je vykresleno samotné hlasování, kde uživatel zvolí z nabídnutých voleb, a poslední strana prezentuje dosažené výsledky v rámci ankety. Avšak datová struktura počítá s libovolným počtem stran a záleží už tedy na konkrétním požadavku. Poté je již snadné prezentaci ankety případně upravit dle žádaného zadání.



Obrázek 6.4: Úvodní obrazovka ankety.

Vykreslení textu, který se na aktuální straně nachází, má na starosti metoda *private void vykresli_stranu(Graphics, int)*. Ta dle informací ze struktury *myRadek* vykreslí všechen text příslušící k dané straně. Dle žádaného typu písma je zvolen vhodný způsob prezentování (např. zda má být užito stínování).

Pro unikátní identifikaci uživatele a znemožnění vícenásobného hlasování je navrženo přihlašování uživatele na základě znalosti jednorázového šesticiferného kódu. To obstarává metoda *private void zaloguj (Graphics)* a vyhodnocení klíče zajišťuje PHP server.

Obrázek 6.5: Přihlašování klienta pomocí číselného kódu.

Grafickou prezentaci aktuální volby uživatele zajišťuje metoda *private void vykresli_sipku(Graphics, int)*, jenž spolupracuje se strukturou *myVolba* a dynamicky vykresluje ukazatel na aktuální volbu dle uživatelova výběru a informací z SQL databáze o pozici voleb.

Jakmile uživatel zvolí svoji odpověď, je odeslána žádost typu *Request* na server. Jako parametry žádosti jsou nastavena data o volbě uživatele a uživateli samotném. To zajišťuje metoda *private void odesliaprijmidata()*.

Obrázek 6.6: Obrazovka volby v rámci ankety.

6.10 Vyhodnocení volby uživatele na PHP serveru

Jakmile uživatel vybere svoji volbu, jsou jeho data odeslána na server a to konkrétně skriptu obsaženému v souboru *zpracuj_volbu.php* a jako parametr jsou přidána data volby a informace o uživateli. Obecné nastavení skriptu a definice přístupu k databázi je stejná jako u souboru *hlavni_data_ankety.php*. Nyní je ověřeno, zda má uživatel na základě svého šestimístného číselného kódu právo hlasovat. Tato informace je uchovávána v databázi v tabulce *seznam_hesel*, kde je ke každému akceptovatelnému šestimístnému kódu přiřazen počet ještě zbývajících možností hlasování. Tato hodnota je v případě proběhnutí hlasování dekrementována o jedna.

Jestliže má uživatel na základě informací z databáze právo hlasovat, je zjištěna jeho volba a ta je zapsána do tabulky s názvem *volby*, která uchovává informace o počtech hlasů a procent jednotlivých možných voleb. Jestliže uživatel nemá právo hlasovat, jeho volba je ignorována.

Do tabulky *sessionmanager* si aplikace ukládá informace o jednotlivých přístupech uživatelů ankety. Uchovává se informace o volbě uživatele, jeho globální i lokální IP adrese, jménu počítače, času přístupu, použitém číselném kódu, a zda byl hlas zapsán.

Následuje zaslání výsledků zpět MHP aplikaci.

ID	volba	global_IP	local_IP	hostname	datum	cas	heslo_uzivatele	platny_hlas
154	2	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:46:24	484314	ne
153	2	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:45:12	123123	ano
152	2	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:38:06	123123	ano
151	2	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:36:51	951427	ne
150	1	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:36:36	123123	ano
149	1	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:35:08	123123	ano
148	2	78.102.115.54	192.168.1.13	PC5	24/04/2009	14:33:53	123123	ano
147	1	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:32:14	123123	ano
146	2	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:26:25	468216	ne
145	7	78.102.115.54	192.168.1.12	myPC	24/04/2009	14:24:28	123123	ano
144	1	78.102.115.54	192.168.1.10	prenosnePC	24/04/2009	14:18:32	123123	ne

Obrázek 6.7: Ukázka tabulky *sessionmanager*.

6.11 Grafická prezentace výsledků ankety

Po odeslání výsledků jednotlivých voleb ze serveru jsou data přijata a následuje vykreslení závěrečné strany s prezentací výsledků ankety.

Výsledky ankety jsou prezentovány formou grafu, což zajišťuje metoda *private void kresli_graf (Graphics g)*. Graf je dynamicky vykreslován v závislosti na počtu voleb a hodnotách počtu procent v nich uložených (ve struktuře *myVolba*).

Obrázek 6.8: Obrazovka volby v rámci ankety.

Při práci s grafikou poskytovanou knihovnami *java.awt.** je nutné respektovat rozdíl prezentace nulového úhlu při vykreslování kruhové výseče. Tato knihovna bere nulový úhel v pravé části kružnice a vzrůstající úhel definuje proti směru hodinových ručiček. Anketa z důvodu divácké přívětivosti prezentuje výsledky s nulovým úhlem v horní části kružnice a s nárůstem úhlu ve směru hodinových ručiček. Rozhraní těchto dvou přístupů zabezpečuje metoda *private int prepocitej_uhel(int).*

7. Závěr

Přechod od analogového vysílání k digitálnímu je zcela neodvratitelný a již nyní je v plném proudu. Tento krok přinese mnoho výhod a nové možnosti, které umožní rozvoj nových služeb, avšak je také spojen s technickými, ekonomickými a právními komplikacemi při jeho zavádění.

Dalším aspektem komplikujícím přechod k této nové technologii je finanční zátěž koncových uživatelů digitálního vysílání, tedy diváků. Zde vyvstává otázka, zda budou diváci ochotni investovat nemalé částky do inovování svého televizního vybavení, které by jim umožnilo zprostředkovat tyto nové služby, např. graficky pojatý teletext, elektronický programový průvodce, videohry, on-line nákupy, hlasování, přístup k internetu a e-mailu. Je důležité, aby televizní a obchodní společnosti přišly s dostatečně poutavými nabídkami praktického využití digitálního vysílání a interaktivních služeb, které přiměje diváky investovat do nové techniky. Tato práce si klade za hlavní cíl právě ono přiblížení problematiky vytváření poutavých aplikací a nástin možných využití v běžném životě.

Praktickou realizací MHP aplikace a jejím popisem je v práci vysvětlen princip vývoje aplikací tohoto standardu a některé jeho možnosti. Na jedné straně je možné využít značné softwarové podpory ze strany dostupných rozšiřujících knihoven, přídavných modulů a nepřeberné škály programů, které mohou být do vývoje těchto aplikací ať už přímo či nepřímo začleněny, avšak na druhé straně nás limitují možnosti hardwaru, tj. výpočetní a paměťové schopnosti set-top boxu a zobrazovací možnosti televize. Jedná se o největší rozdíl, který odlišuje tyto aplikace od vývoje např. počítačových aplikací. Tato práce uvádí přístup, jak je možné se těmto odlišnostem (např. životní cyklus, omezení paměti a rychlosti) přizpůsobit.

S digitalizací spojená masivně rozšířená a plně interaktivní platforma MHP je v naší zemi prozatím jen budoucností a čas pro její největší rozvoj teprve nastane, avšak již nyní je nutné připravovat takovéto aplikace, které budou pro koncového diváka prospěšné a poutavé.

8. Použitá literatura

[1] LEGÍŇ, M. *Televizní technika DVB-T*, Praha: BEN, 2006. 286 s. ISBN 80-7300-204-3

[2] ŘÍČNÝ, V., KRATOCHVÍL, T. *Základy televizní techniky*, Brno: Ústav radioelektroniky, 2006, 161 s. ISBN 80-214-3203-9

[3] MORRIS, S., CHAIGNEAU, A. *Interactive TV standards*, Amsterdam: Elsevier. 2005. 585 s. ISBN 0-240-80666-2

[4] SCHWALB, E. M. *ITV Handbook: Technologies and Standards*, London: Prentice Hall, 2004. 723 s. ISBN 0-13-100312-7

[5] REIMERS. U. *DVB – The Family of International Standards for Digital Video Broadcasting*, Berlin: Springer, 2005. 408 s. ISBN 3-540-43545

[6] LUGMAYR, A. *Digital Interactive TV and Metadata*, Berlin: Springer, 2004. 254 s. ISBN 0-387-20843-7

[7] DVD Project Office. *The Open Standard for Interactive Television*, Poslední aktualizace 07. 04. 2009. Dostupné z WWW: <http://www.mhp.org/>

[8] E-Studio s.r.o. *Digitální televize.* Poslední aktualizace z roku 2009. Dostupné z WWW: <www.digitalnitelevize.cz>

[9] Wikipedia. *Xtea cipher*. Poslední aktualizace 13. 04. 2009. Dostupné z WWW: < http://en.wikipedia.org/wiki/XTEA>

[10] Loch, J. *Effective Java*, Upper Sadle River: Addison-Wesley, 2008. 348 s. ISBN 0-321-35668-3

[11] Dobda, L. *Ochrana dat v informačních systémech*, Praha: Grada, 1998. 286 s. ISBN 80-7169-479-7

[12] Doseděl T. *Počítačová bezpečnost a ochrana dat*, Brno: Computer Press, 2004. 190 s. ISBN 80-251-0106-1

[13] Burda K. *Bezpečnost informačních systémů*, Brno: FEKT VUT Brno, 2005, 104 s.

9. Seznam použitých zkratek

API	Application Programming Interface
AES	Advanced Encryption Standard
AIT	Application Information Table
AoD	Audio on Demand
AWT	Abstract Window Toolkit
DES	Data Encryption Standard
DVB	Digital Video Broadcasting
DVB-T	Digital Video Broadcasting - Terrestrial
EPG	Electronic Program Guide
GSM	Gobal System for Mobile Communications
HAVi	Home Audio / Video Interoperability
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IPSec	Internet Protocol Security
JDK	Java Development Kit
JVM	Java Virtual Machine
MHP	Multimedia Home Platform
MPEG	Motion Picture Experts Group
OFDM	Orthogonal Frequency Division Multiplexing
PHP	Personal Homepage
PPV	Pay per View
RSA	Rivest, Shamir, Adleman
SFN	Single Frequency Network
SQL	Structured Query Language
SSL	Secure Sockets Layer
TEA	Tiny Encryption Algorithm
UMTS	Universal Mobile Telecommunications System
UTF	Unicode Transformation Format
VoD	Video on Demand
WEP	Wired Equivalent Privacy
XTEA	eXtended Tiny Encryption Algorithm