Autentisering via GSM.

LUNDS
UNIVERSITET
Lunds Tekniska Högskola

LTH Ingenjörshögskolan vid Campus Helsingborg
Datateknik

Examensarbete:
Mathias Nordin

LTH Ingenjörshögskolan vid Campus Helsingborg
Lunds universitet
Box 882
251 08 Helsingborg

LTH School of Engineering
Lund University
Box 882
SE-251 08 Helsingborg
Sweden

Tryckt i Sverige
Media-Tryck
Biblioteksdirektionen
Lunds universitet
Lund 2012

Sammanfattning

Examensarbetet är en påbyggnad av en befintlig autentiseringstjänst som tillhandahålls av Crunchfish AB och Ubiqo security AB. Autentiseringstjänsten bygger på att en användare utnyttjar en personlig mobiltelefon och dess internetuppkoppling för att bekräfta sin identitet på internet. Företagen är intresserade av att utöka tjänsten så att autentiseringen kan göras via mobilens GSM-nät. Detta examensarbete handlar om att utveckla en första prototyp av denna GSM-funktion. Examensarbetet levereras delvis i denna rapport samt delvis i kod för server och mobiltelefon. Resultatet blev en prototyp som kan visas för kunder och investerare. I rapporten framgår vilka delar av systemet som behöver utecklas för storskalig användning av tjänsten, samt delar som ger ökad pålitlighet och lönsamhet.

Nyckelord: Autentisering, Säkerhet, GSM, Android

Abstract

The thesis is an extension of an existing authentication service provided by the companies Crunchfish AB and Ubiqo security AB. The authentication service is based on a personal mobile phone and it's internet connection to verify his/her identity on internet. The companies are interested in expanding the service so that authentication can be done through the mobile phone's GSM network. This thesis is about developing a first prototype of this GSM-function. A part of the report is delivered in this report and the other part is delivered in code for the server and mobile phone. The result was a prototype of the service to be used for presentations for costumers and business partners. The report explains parts of the system that needs to be develope before large-scale implementation of the system to increased reliability and profitability.

Keywords: Authentication, Security, GSM, Android

Förord

Jag vill tacka Ben Smeets professor på LTH för intressanta diskussioner samt vägledning under examensarbetet. Jag vill tacka Christian Nyberg som hjälpt till med formailia och uppbyggnad av denna rapport. Jag vill även tacka alla på Crunchfish i Malmö för trevligt sällskap samt gratis kaffe.

Innehållsförtäckning

1 Inledning

Syftet med detta examensarbete är att analysera en utökning av tillgängligheten till GSM-nätet för en specifik autentiseringstjänst. Detta skall göras med målet att systemet skall vara så flexibelt, ekonomiskt, användarvänligt och säkert som möjligt.Inledningen innehåller de väsentliga delar som behövs specificeras innan man kan förklara examensarbetet.

1.1 Bakgrund

Examensarbetet utvecklades i samverkan med företaget Crunchfish[1] i Malmö.Crunchfish är ett företag som är specialiserade på innovativa lösningar och jobbar mycket med mobilapplikationer. Ett av Crunchfishs sammarbetspartner är Ubiqo[2] security AB som säljer egenutvecklade säkerhetslösningar för banktransaktioner och för autentisering på internet. Examensarbetet skall analysera en utökning av tillgängligheten för autentiseringstjänsten. Händelseförloppet för den befintliga autentiseringstjänsten är att en användare skriver in ett användarnamn på en hemsida och trycker på knappen logga in. Sedan dyker en förfrågan upp på användarens mobiltelefon som undrar om användaren godkänner inloggningen. Användaren besvarar frågan och beroende på val loggas användaren in eller inte. Autentiseringstjänsten använder sig idag av mobiltelefonens internetanslutning. Problematiken med en telefons internetanslutning är att signalstyrkan är dålig på vissa ställen och det är ofta väldigt dyrt att vara ansluten till internet när man är utomlands. Crunchfish är intresserade av att utöka tillgängligheten av autentiseringstjänsten genom att utnyttja GSM-nätet. Detta skulle även göra tjänsten billigare för vissa användare som vistas utomlands. Det är viktigt att examensarbetets reultat är så likt den autentiseringstjänst som är utvecklad som möjligt eftersom man inte vill att användargränsnittet skall påverkas.

1.2 Syfte

Examensarbetet skall granska lösningar för att autentisera sig på internet via en mobiltelefons GSM-nät. Man skall även försöka göra en prototyp som bevisar att en sådan uppkoppling är möjlig. Man skall sträva efter att tjänsten skall vara så ekonomisk, flexibel, användarvänlig och säker som möjligt. Det skall även skrivas en rapport som innehåller information om systemet.

1.3 Målsättning

Examensarbetet har följande mål:

1. Att utreda hur server och mobiltelefon skall kommunicera via GSM.
2. En egenutvecklad prototyp för verifiering av mobiltelefonen via GSM som skall kallas via ett funktionsanrop i PHP.
3. Mobiltelefonen skall automatiskt fråga användaren om han eller hon vill acceptera inloggningen när en förbindelse skapas med servern.
4. De delar av systemet som skickar ut och tar emot SMS skall vara oberoende av vilken mobiltelefon användaren har. Detta gör att delar av systemet kan återanvändas när man skall göra tjänsten tillgänglig för flera mobiltelefoner.

1.4 Problemformulering

De följande tre frågorna var de viktigaste att besvara:

1. Är det möjligt att via GSM-nätet utöka Ubiqos autentiseringstjänst på ett användarvänligt, billigt och säkert sätt?
2. Vad är den ungerfärliga kostnaden för att autentisera sig på det valda autentiseringstjänsten?
3. Vad skall man ta hänsyn till vid storskalig implementering av systemet?

1.5 Avgränsningar

Den prototyp som skall göras skall vara utvecklad för operativsystemet Android. De delar av prototypen som skall implementeras på servern skall utvecklas på en testserver, annars kan man störa den verkliga server som är i drift. Prototypen skall inte behandla komplikationer vid fleranrop av tjänsten. Prototypen skall vara utvecklat med lönsamhet, flexibilitet och säkerhet som huvudsakliga kriterier att uppfylla. Med en flexibel uppkoppling menas en uppkoppling som är kompatibel med flera olika mobiltelefoner. Prototypen skall inte integreras med det redan utvecklade systemet, eftersom det är begränsad tid avsatt för examensarbetet.

1.6 Liknande arbeten

Det finns liknande arbeten som har gjorts tidigare inom autentisering via mobiltelefon. Secure Web Authentication with Mobile Phones är ett examensarbete gjort av Min Wu, Simson Garfinkel och Rob Miller på MIT Computer Science and Artificial Intelligence Laboratory Cambridge [3]. Arbetet handlar om att säkerställa autentisering på internet via en mobiltelefon. Där ingår kommunikation via SMS vilket är en tjänst inom GSM-kommunikation. Arbetet är mer allmänt och har inget specifikt företag att utveckla för. Detta gör att frågor såsom ekonomi och flexablitet kommer i bakgrunden. Secure Web Authentication with Mobile Phones är ett försök till

standard inom autentisering på internet, det är inget system som skall implementeras och installeras hos kunder.
Det har även gjorts ett arbete om att använda mobila enheter till autentisering via internet. Arbetet heter using a Personal Device to Strengthen Password Authentication from an Untrusted Computer och är skrivet av Mohammad Mannan samt P. C. van Oorschot på School of Computer Science Carleton University, Ottawa, Canada[4]. Arbetet är likt Secure Web Authentication with Mobile Phones men är mer inriktat mot säkerhet och matematiska bevis av säkerheten. Detta system finns som referens i Secure Web Authentication with Mobile Phones. Systemet ska ge ett säkert sätt att logga in med en portabel enhet och innehåller kommunikation mellan server, hemsida, mobiltelefeon samt användare. Komunikationen mellan telefon och webbrowser är i detta fallet SMS.

1.7 Källkritik

De grundläggande kunskaperna inom examensarbetet erhölls från publicerade böcker som används som läromedel inom högre utbildning, såsom [11]. Djupare kunskaper erhölls från kända internetsidor såsom [12] och [19]. Dessa sidor används av många programmerare vid inlärning och [12] innehåller bland annat mycket information om standardbibliotek. De källor som kan vara diskutabla är företagssidor såsom [9]. Men författaren anser dessa källor som pålitliga eftersom de fungerar som support mot företagens kunder. Om informationen hade varit felaktig hade inte företagets tjänster fungerat. Eftersom företagen har flera betalande kunder kan man därför anse att informationen är korrekt. Liknande gäller koder tagna från forum såsom[17]. Efter implementering kan man se att koden fungerar.

2 Teknisk bakgrund

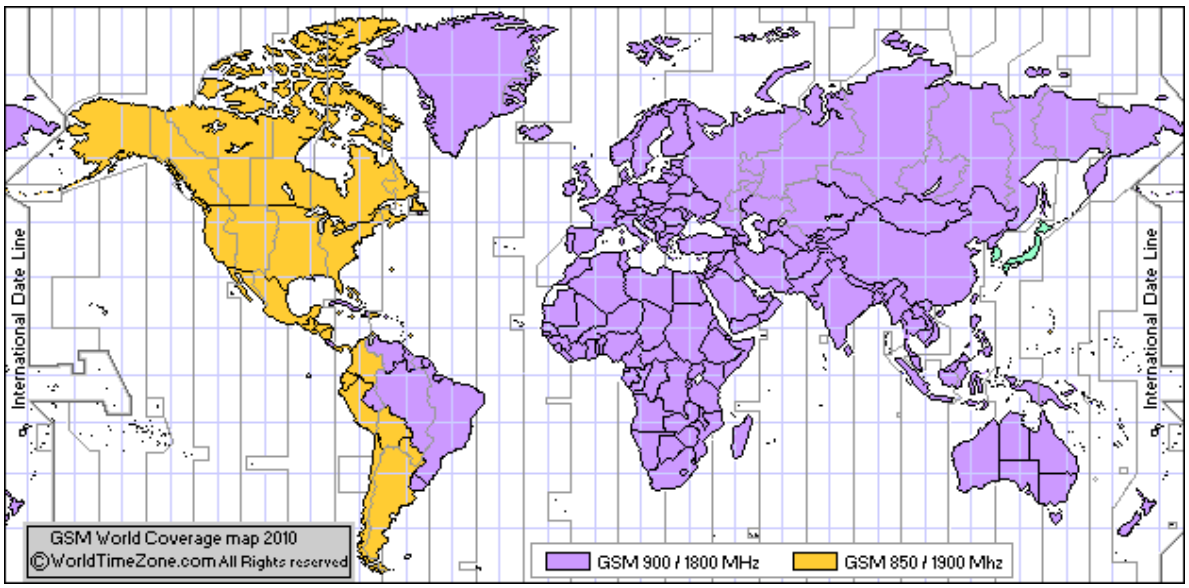
Här presenteras den teknik som har använts för att implementera prototypen.

2.1 GSM

GSM är världens största kommunikationsteknologi för mobiltelefoner som sammanlagt förbinder över 4 miljarder enheter. Fördelen med att använda GSM istället för enbart internet är att GSM har större täckning än mobilt internet. För en del användare kommer detta att vara ett billigare alternativ vid autentisering eftersom mobiltelefonerna slipper ha en ständig uppkoppling till internet.

2.1.1 Historia

År 1982 började Conférence Européene des Postes et Télécommunications (CEPT) specificera det som en dag skall kallas GSM[5]. CEPT är en organisation bestående av representanter från olika europeiska länder som diskuterar analog/digital kommunikation för att kunna skapa internationella standarder. 1986 allokerades 900MHz-bandet för GSM och senare 1800MHz-bandet inom Europa. 2001 överskred antalet abonnenter 500 miljoner. GSM innehåller flera tjänster, bland annat SMS, tal och MMS. Den 31 december år 2006 fanns GSM tillgängligt för 97-98% av Sveriges befolkning. I figur 1 visas de länder som har GSM tillgängligt.



Figur 1. Bilden är tagen från: http://www.worldtimezone.com/gsm.html

2.1.2 GSM kryptering

Eftersom GSM-nätet används när man skall autentisera sig mot servern, utnyttjas de säkerhetsmekanismer som GSM redan erbjuder. GSM har idag två olika krypteringar som är aktiva och en som är utvecklad men inte etablerad. Anledningen till att det finns olika krypteringar är att GSM är en relativt gammal standard.

De krypteringsalgoritmer som kan användas är A5/1, A5/2 och A5/3. Det är viktigt att påpeka att dessa standarder anger hur trafiken mellan basstation och mobil ska krypteras. Skyddet av informationen från en basstation och vidare i nätverket är mobiloperatörens ansvar. Detta betyder att man kan anta att data som skickas är oskyddad för läsning av obehöriga när den färdas från sändare till mottagare.

2.1.2.1 A5/1

A5/1 är den krypteringsalgoritm som först implementerades och som i dagens läge är den säkraste i användning inom GSM. Dock har den blivit knäckt av Alex Biryukov, Adi Shamir and David Wagner (2000)[6]. Metoderna har sedan förfinats och idag är den effektivaste utvecklad av en tysk datoringenjör vid namn Karsten Nohl[7]. A5/1 kryptering går ut på att man skapar ett strömchiffer hos operatören och ett strömchiffer hos abonnentens telefon, med hjälp av den 64 bitars hemliga nyckel som finns hos operatören samt i abonnentens SIM-kort. Med detta strömskiffer som finns både hos leverantören och abonnenten kan man dekryptera samt kryptera GSM-kommunikationen mellan de båda parterna.
Karsten Nohl testade olika sätt att knäcka detta strömmchiffer i sin avhandling från Security Research Labs i Berlin. Han påvisade att det lättaste sättet att knäcka krypteringen var att använda en kombination av rainbowattack och distinguished points. Distingushed points är delar av det krypterade meddelandet där man vet klartextens innehåll. I detta fallet visste man att i slutet av det krypterade meddelandet fanns nollor krypterade och genom att lagra dessa kan man finna likheter och på så sätt knäcka krypteringen. Rainbowattack är ett sätt att knäcka strömchiffer som kräver mindre CPU men större mängd minne lagrat i form av en rainbowtabell. En Rainbowtabell är ett sätt att lagra data för att lättare knäcka hashfunktioner. Karsten Nohl använde en rainbowtablell på 2TB som är lagrad på två SSD diskar. Att göra själva tabellen med hans algoritm tog cirka en månad. När väl tabellen är gjord kan han knäcka A5/1 kryptering på cirka 5 sekunder om han använder två kraftfulla grafikkort.

2.1.2.2 A5/2

A5/1 kunde inte exporteras till östra Europa på grund av att exportrestrektioner saknades. Därför utvecklades den svagare A5/2-krypteringen som inte hade sådana restriktioner.
A5/2 är ett strömchiffer för kryptering som utvecklades av SAGE gruppen under ETSI (European Telecommunications Standards Institute) för att användas i östra Europa där nödvändiga tillstånd inte fanns tillgängliga för att använda A5/1-kryptering. Dock påvisades denna kryptering vara väldigt svag när Ian Goldberg och David Wagner samma månad som den började användas knäckte den. [8] De visade att detta gick att göra med enkel utrustning inom

rimlig tid. Därför implementeras inte idag A5/2 kryptering i GSM-nät och avvecklas i länder som använder det.

2.1.2.3 A5/3

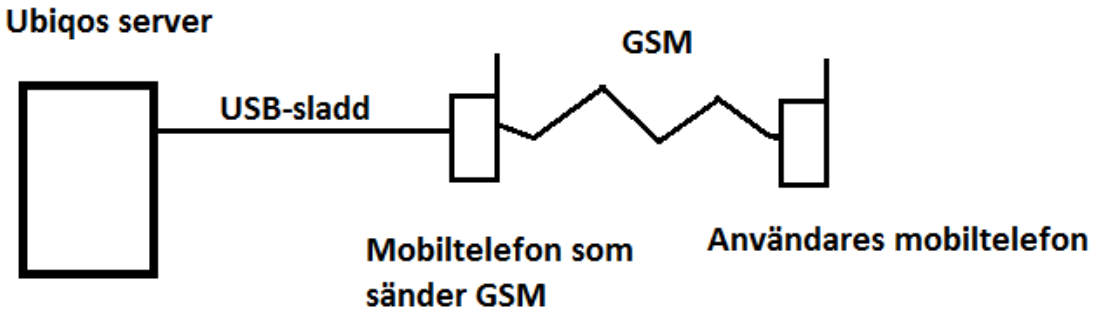
A5/3 är en strömchifferkryptering utvecklad av SAGE gruppen. Den bygger på ett blockchiffer som man gör om till ett strömchiffer. Den finns även i en variant med 128-bitarsnyckel under beteckningen A5/4.

2.2 Sändning av GSM

Det är två viktiga vägval som måste göras innan man kan sända GSM. Det första är att bestämma hur man skall sända GSM och det andra är att bestämma vilket medium man skall sända det i. Detta kapitel innehåller fakta som ligger till grund för dessa beslut.

2.2.1 Uppkoppling

Det skall etableras en förbindelse mellan Ubiqos server och en mobiltelefon. Detta går att göra på flera olika sätt och här nedan beskrivs de som har studerats under examensarbetet. Via PHP skall en server kommunicera med en sändare som med hjälp av ett SIM-kort sänder GSM till en mobiltelefon. SIM-kortet behövs eftersom det är det som identifierar en abonnent i GSM-nätet. En nackdel med GSM är att tjänsten kan bli överbelastad vid större högtider såsom nyårsafton. Denna nackdel gäller även kommunikation via internet.



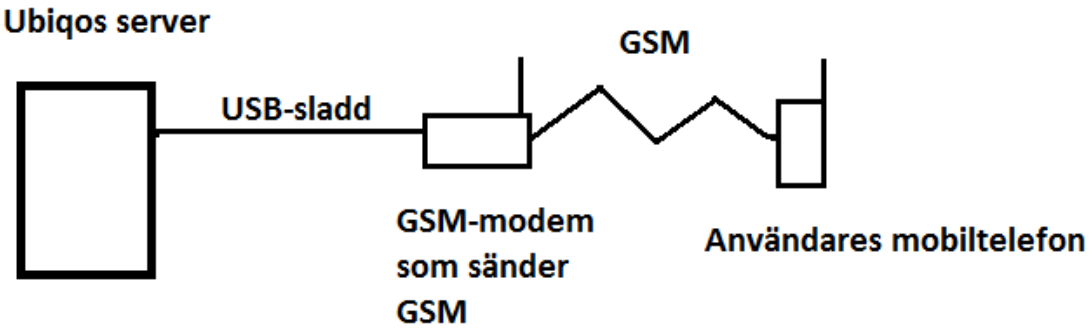
Figur 2. Visar en möjlig uppkoppling för kommunikation via GSM.

I figur 2 visas en förbindelse som inte är vanlig och svår att hitta information om. Detta kan bero på flera faktorer.

1. Man måste programmera de båda telefonerna och det är främst på senare år det har blivit möjligt för privatpersoner att programmera mjukvara för mobiltelefoner.
2. Mobiltelefoner har tidigare varit mer begränsade teknikmässigt i jämförelse med dagens mobiltelefoner.

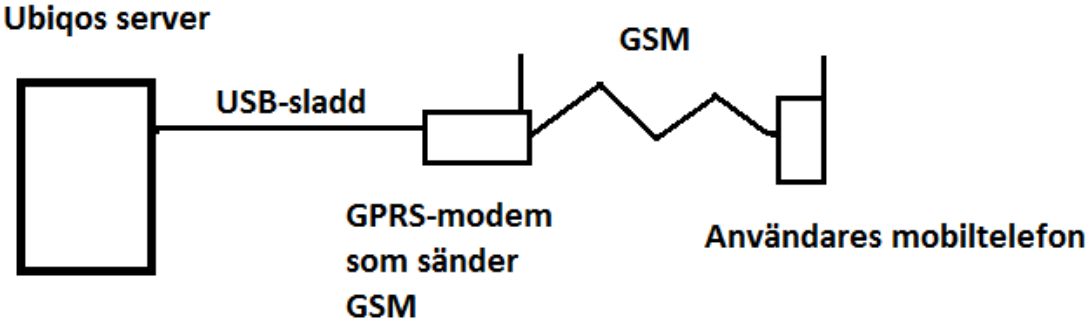
3. Osäkert ifall det kommer ge bra prestanda när GSM-trafiken mot telefonen ökar.

Fördelen med denna uppkoppling är att ingenting behövs inköpas för att förbindelsen skall fungera eftersom författaren kan använda sin egen mobiltelefon vid implementering av systemet.



Figur 3. Visar en möjlig uppkoppling för kommunikation via GSM.

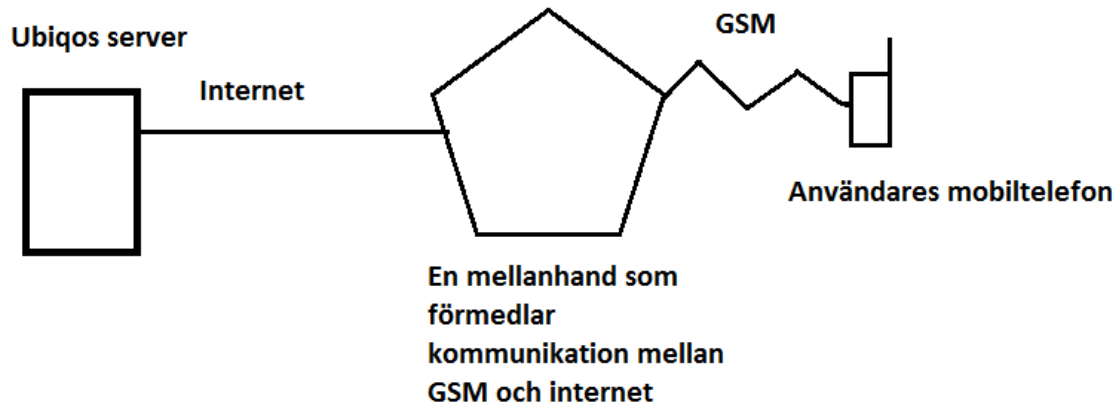
I figur 3 kopplas servern upp med hjälp av ett GSM-modem[9]. Ett GSM-modem kan betraktas som en mobiltelefon fast den är utformad för datorer och servrar. Om man skall mäta GSM-modems ungerfärliga kapacitet gällande skicka och ta emot SMS klarar den av att behandla ungefär 10SMS/min och kostar runt 1665kr. Detta är relativt dyrt om man jämför med uppkopplingen som visas i figur 4.[9]



Figur 4. Visar en möjlig uppkoppling för kommunikation via gsm.

Uppkopplingen ser nästan likadan ut som figur 3, dock med skillnaden att ett GPRS-modem används istället för ett GSM-modem. Ett GPRS -modem klarar av cirka 30SMS/min. GPRS står för General Packet Radio Service och är en utökning av GSM. Skillnaden mellan GSM och GPRS är att man har utnyttjat mer än en tidslucka vid TDMA (Time Division Multiple Access)[10]. Tack vare detta sätt att läsa och skicka signalerna kommer man upp i högre hastigheter. Skillnaden på pris mellan ett GPRS-modem i jämförelse med ett GSM-modem

är väldigt liten i jämförelse med att ett GPRS-modem har cirka 3 gånger så hög kapacitet som ett GSM-modem.



Figur 5. Visar en möjlig uppkoppling för kommunikation via gsm.

I figur 5 visas en lösning som endast kan behandla SMS och MMS. Alltså ges inte möjlighet att kommunicera med hjälp av tal inom GSM. Priserna för detta beror på vilken mellanhand som används. Det är även skillnad på mellanhändernas kapaciet. Fördelen med denna form av sändning är att kommunikationen är lätt och snabb att implementera då man får tillgång till företagets API. Man kan även knyta samman flera mellanhänder för att kunna skicka och ta emot SMS/MMS från flera delar av världen. Detta är bra eftersom man då kan ha en mellanhand från varje land och på så sätt undvika att SMS/MMS skickas länder emellan. Om SMS/MMS skickas mellan länder är det stor risk att SMSen eller MMSen blir dyrare att skicka.

2.2.2 Medium

För att kunna sända viktig information via GSM måste man bestämma vilka medium man skall sända genom. Olika medium kan vara SMS, MMS och vanligt tal. De krav som ställs på mediumet är att det skall vara snabbt, billigt samt relativt enkelt kunna kommunicera med en mobiltelefon. Det vore bra om man kan kryptera informationen som transporteras inom mediumet, vilket leder till att man får krypterad information om någon kommer förbi mobiloperatörernas säkerhet.

2.2.2.1 SMS

SMS står för Short Messege Service och är små texter (160 tecken) som skickas mellan telefoner. SMS är ett väldigt utbrett medium för kommunikation mellan servrar och telefoner. Det förekommer bland annat vid reklam och marknadsundersökningar. Eftersom SMS redan är utbrett som medium vid kommunikation mellan server och mobiltelefon leder det till att det finns mycket dokumentation att tillgå. Priset för att skicka ett SMS varierar beroende på mobioperatör, abonnemang samt vilken slags och hur lång text som skickas i SMSet. Eftersom en så låg kostnad för användare som möjligt är önskvärd skall det skickas max 160 tecken i ett SMS med tecken som ingår i

GSM 7-bit alfabetet. GSM 7-bit alfabetet är utvalda tecken ur unicodes alfabet. Om man skulle mata in ett tecken som är utanför GSM 7-bit alfabetet så utökas tecknens längd till unicode 16-bit vilket gör att endast 70 tecken kan skickas. För att hålla en god form av säkerhet behöver innehållet i SMSet vara krypterat så att ingen utomstående skall kunna läsa känslig information som skickas i SMSet. Ett typiskt pris för ett SMS med kontantkort är 0.59 SEK (Telenor). En nackdel med SMS är att du inte kan vara säker på att ett SMS har kommit fram till en abonnent. Det finns ytterligare tjänster som möjliggör detta, men det tillkommer ofta en extra summa.

2.2.2.2 MMS

MMS står för Multimedia Messaging Service och är en vidareutveckling av SMS. MMS har samma egenskaper beträffande kryptering, snabbhet och tillgänglighet som SMS, men skillnaden är att ett MMS-meddelande kan innehålla mer information och kostar mer. Detta resulterar i att MMS kan skicka enkla bilder och filmer samt skicka tecken. Fördelen med MMS är att man kan skicka 1024 tecken, men detta resulterar också i ett högre pris. Samma kontantkort som angavs i 2.2.2.1 kostade ett MMS 1.59 SEK att skicka.

2.2.2.3 Tal

Det är svårt att hitta information angående denna typ av medium. Detta beror förmodligen på att det är svårt att skicka information via talområdet så att en server skall förstå meddelandet. Att implementera kommunikation via talets bandbredd hade förmodligen kunnat lösas, men det skulle förmodligen ta för lång tid från resten av examensarbetet.

2.3 PHP

PHP är en väsentlig del i examensarbetet eftersom servern som skall anropa funktionen är en LAMP-server. LAMP står för Linux Apache Mysql PHP/Python/Pearl. Linux är operativsystemet servern bygger på. Apache är det bibliotek av öppen källkod som systemet bygger på. Mysql är den databas som finns tillgänglig på servern. PHP,Python och Pearl är de skriptspråk som kan exekveras på servern[11]. Det färdiga systemet är skrivet i PHP och därför skall även koden som produceras under examensarbetet vara skrivet i PHP. PHP är en akronym för Hypertext Preprocessor. PHP är ett skriptspråk som ofta används inom webbutveckling och är instruktioner som exekveras på en server. Detta skall inte förväxlas med exempelvis JavaScript som exekveras på den dator som kommunicerar med servern. PHP ger möjligheter att bland annat hämta/lagra värden ur serverns databas, använda apaches bibliotek så man exempelvis kan behandla pdf-filer på olika sätt och mer därtill. PHP är även väldigt flexibelt och kan implementeras tillsammans med HTML-kod. Här nedan visas ett exempel.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
    "http://www.w3.org/TR/html4/loose.dtd">
<html>
    <head>
        <title>Example</title>
    </head>
    <body>

        <?php
            echo "Hi, I'm a PHP script!";
        ?>

    </body>
</html>
```

Figur 6. Koden är tagen från php.net http://se.php.net/manual/en/intro-whatis.php.

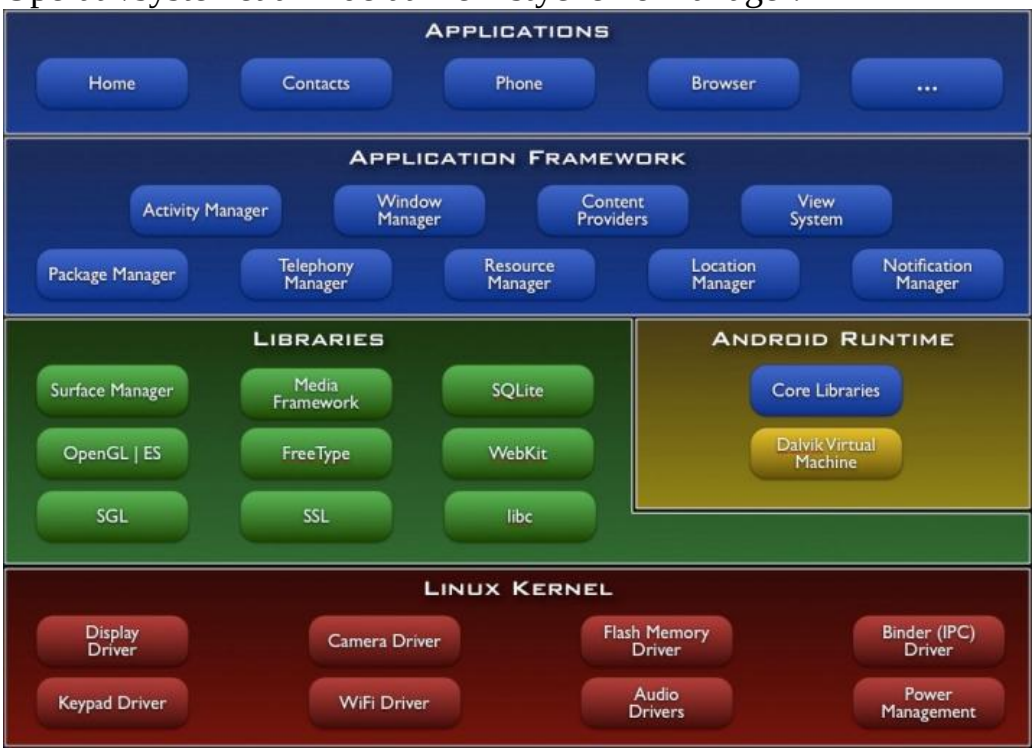
Det som är skrivet inom <?php ?> är php-kod. Som man ser är detta implementerat bland HTML-kod som visar en väldigt enkel hemsida. Det denna PHP-koden gör är att skriva ut texten Hi, I´m a PHP script!.

2.4 Android

Eftersom kommunikationen mellan servern och telefonen skall vara krypterad behöver telefonen mjukvara som kan dekryptera inkommande data samt kryptera utgående data. Det behövs även mjukvara för att kommunicera med användaren på ett användarvänligt sätt. Android är ett operativsystem som är ursprungligen utvecklat av Android Inc som köptes up av Google 2005[12]. Utvecklingen av Android sker av Open Handest Alliance som består av flera stora företag.

2.5 Uppbyggnad

Operativsystemet är indelat i fem stycken olika lager.



Figur 7. Bilden är tagen från http://mer.android.com/guide/basics/what-is-android.html

2.5.1 Applikations lagret

I applikationslagret finns applikationerna. Med applikationer menas de mjukvaruprogram som är installerade på telefonen. Applikationerna använder funktioner från de underliggande lagrena inom Androids operativsystem. Alla applikationer är programmerade i Java.
Det är här som examensarbetets applikation kommer vara installerad.

2.5.2 Applikationsramverket

Applikationsramverket är verktygen som applikationerna använder sig av för att utföra sina funktioner. Alla applikationer på en Andoid har tillgång till samma ramverk.
De funktioner som kommer att användas vid utvecklingen av deltjänsten tillhör telephony manager och ger bland annat möjligheten att läsa SMS. applikationen behöver ha tillgång till.

2.5.3 Bibliotek

Biblioteket består av verktyg som applikationsramverket använder sig av. Exempel är SQLite som är en databas, samt mediaramverket som har olika codex till videofiler, bildfiler och musikfiler etc. Detta är ingenting som applikationen behöver ha tillgång till.

2.5.4 Android runtime

Består av två olika delar.
1 Core Libraries. Är mycket av de standardbibliotek som finns tillgängligt i Java.

2. Dalvik Virtual Machine. När en applikation skall exekvera sin kod skapas en instans av Dalvik Virtual Machine. Den är här koden exekveras specifikt för varje applikation.

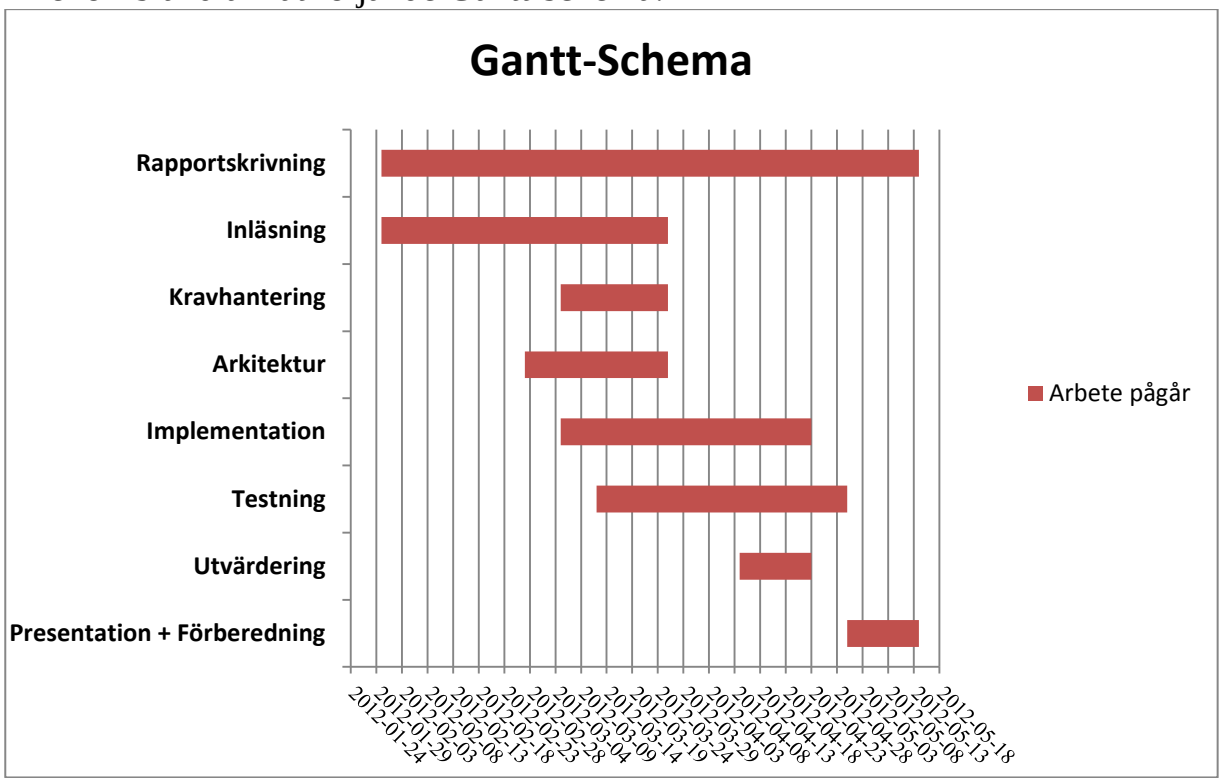
2.5.5 Linuxkärna

Linux är kärnan som operativsystemet bygger på. Det är den som sammanför mjukvara med hårdvara samt minneshantering, grundläggande säkerhet m.m. Självfallet använder applikationen Linuxkärnan men det är ingenting som man behöver tänka på under utvecklingen.

3 Metod

3.1 Tiden

Till en början gjordes en specifikation av examensarbetet för att specificera examensarbetet och förenkla utvecklingsprocessen. Denna specifikation innehöll bland annat följande Gantt-schema.



Figur 8. Detta Gantt-schema fanns med i examensarbetets specificering.

Man kan konstatera att detta Gantt-schema följdes väldigt dåligt under projektet. Detta berodde på att det tog lång tid att komma fram till vad som skulle utvecklas inom Ubiqos inloggningstjänst.

3.2 Utvecklingsmetod

Eftersom det var ospecificerat vad som skulle utvecklas under examensarbetet valdes en iterativ metod att jobba efter. Det var endast en person som jobbade med projektet så författaren valde ingen känd iterativ metod såsom Kanban eller Scrum. Författaren ansåg att det var onödig att ha projekttavlor och möten då endast en person skulle använda dessa. Den utvecklingsmetod som användes under projektet var iterativ och bestod ofta av notiser innehållande framtida planering. De huvudsakliga anledningarna till detta var svårighet vid implementering eller tidsbrist.

3.3 Insamling av information

Man kan dela in informationsinsamlingen i två olika kategorier. Till en början bestod informationsinsamlingen i princip endast av djupgående information.

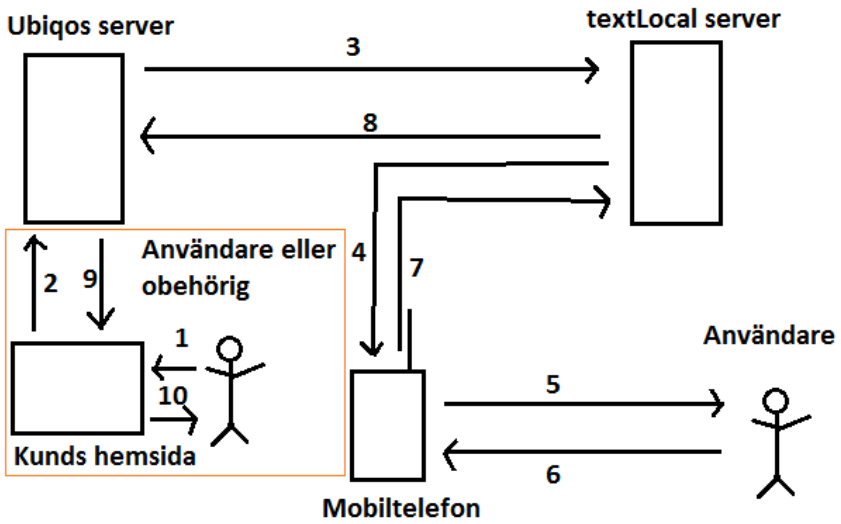
En bok om PHP[11] lästes och en hemsida [12] användas för att författaren skulle lära sig Androidutveckling. Mycket tid gick även till inlärning av GSM. Senare övergavs denna djupgående insamling av information eftersom det inte fanns tid att bygga upp hela systemet från grunden. Istället hämtades olika delar av systemet från olika sidor på internet för att sammanfogas till resultatet av detta examensarbete. Fördelen med detta är att man snabbt kan få resultat, men nackdelen är då att man ofta förlorar djupgående kunskap.

4 Analys

Här beskrivs analysen av prototypen.

4.1 Arkitektur

Efter informationshämtningen kunde den övergripande arkitekturen för systemet skissas.



Figur 9. En övergripande bild av arkitekturen av prototypen.

Det som är markerat inom den orangea rutan ingår inte i examensarbetet. Detta är redan färdigutvecklat och finns endast med i bilden för att ge en tydligare bild hur systemet fungerar.

Såhär fungerar systemet vid en autentisering:
Kund = ett företag som har använder Ubiqos autentiseringstjänst.

1. En användare eller obehörig försöker logga in på en kunds hemsida. Detta görs genom att skriva in ett användarnamn på hemsidan. Hemsidan finns installerad på kundens server eller Ubiqos server.
2. Kundens server säger till Ubiqos server att denna användare försöker logga in på deras hemsida.
3. Ubiqos server anropar en funktion och ger funktionen ett telefonnummer samt en hemlighet som indata. Denna hemlighet krypteras och skickas tillsammans med användarens telefonnummer till textLocal.
   Denna förbindelse är krypterad med SSL.
4. TextLocal skickar det krypterade meddelandet till det telefonnummer som angavs från Ubiqos server.
5. Applikationen märker att den har fått ett meddelande från textLocal och efter dekryptering av meddelandet märker den att det är rätt hemlighet i meddelandet. Applikationen visar användaren ett alternativ att antingen acceptera inloggningen eller förneka den.
6. Användaren svarar på frågan.

7. Om användaren valde att tillåta inloggningen på hemsidan skickas ett krypterat SMS till textLocal. Om användaren inte ville acceptera inloggningen skickades inget SMS till textLocal för att spara en onödig utgift.
8. TextLocal vidarebefodrar informationen till kundens server.
9. Servern behandlar informationen som erhålls från textLocal och kan därefter verifiera ifall det är rätt telefon som har skickat meddelandet. Om så är fallet beviljar servern tillträde till hemsidan.
10. Användaren loggas in på hemsidan.

4.2 Servern

Eftersom man inte ville störa Ubiqos server som är i drift så implementerades PHPfunktionen på en testserver. Testservern är en WAMP (Windows Apache MySQL PHP/Python/Pearl) server vilket är snarlik LAMP men skillnaden att operativsystemet är Windows istället för Linux. Funktionens huvudsakliga uppgifter är att skicka och ta emot information från Ubiqos api samt verifiera att det är rätt telefon man kommunicerar med. Eftersom inget svar skickas från användarens mobiltelefon om användaren väljer att neka inloggningen borde funktionen returnera false (stängas av) när en viss tid har passerats. Detta har dock inte hunnits implementerats i examensarbetet eftersom det är en dålig lösning att servern skall starta en timer själv. Det vore bättre att integrera en timer på den hemsida användaren kopplar upp sig emot eftersom JavaScript är bättre lämpat för detta.

4.3 SMS-leverantör

Som kommunikationsmedium valdes SMS. Detta valdes för att det är en beprövad och väldokumenterad form av kommunikation mellan server och mobiltelefon. För att skicka SMS valdes kommunikationen som förlitar sig på en mellanhand och visas i figur 5. Anlednigen till att denna metod valdes var att den är enkel att implementera och är lätt att vidareutveckla i större skala. Implementeringen är även väldigt billig i början då ingen hårdvara behövs köpas in eftersom författaren har en egen WAMP server. Nackdelen med denna implementering är att man är beroende av att leverantören har en bra säkerhet. Bland annat kopplar servern upp sig mot mellanhanden med ett användarnamn och lösenord. Dock kan man göra både användarnamnen och lösenorden väldigt säkra då endast servern behöver logga in på kontot. Det kommer med all säkerhet inte gå att knäcka lösenordet med ordlista attacker eller liknande, utan endast med att fiska efter lösenordet eller väldigt tidskrävande brute force attacker. Om det blir många användare av tjänsten vore det även bra att ta kontakt med leverantören om man kan göra inloggningen mot deras server på ett ännu säkrare sätt.

4.3.1 Olika SMS-leverantörer

Det finns ett flertal olika leverantörer som ger möjlighet att skicka och ta emot SMS. Det är svårt att veta vilken leverantör som är bäst rent ekonomiskt för Crunshfish och Ubiqo eftersom man ännu inte vet vilka framtida kunder företagen kommer att ha. Har man en kund med de flesta av sina användare i England är en leverantör från England det billigaste alternativet. Här nedan finns de två leverantörer som jämfördes under detta examensarbete.

4.3.2 TextLocal

TextLocal är en brittisk leverantör som har stora kunder såsom Renault samt mindre kunder såsom lokala restauranger. Fördelen med textLocal är att det är lätt att registrera sig och komma igång med tjänsten. De ger även möjlighet att koppla upp sig mot deras server via en https förbindelse. En https förbindelse är en förbindelse med en TLS- eller SSL-kryptering. Både TLS och SSL anses vara bra krypteringar och används ofta vid banktransaktioner. Det kostar 0.65 kronor per SMS att skicka SMS till Sverige men priserna kan sjunka om man beställer större kvantitet av SMS. Exempelvis kostar det 0.47 kronor att skicka ett SMS till Sverige om man köper 50 000 SMS. För att textLocal skall kunna ta emot SMS från mobiltelefoner kostar det 398 kr i månaden exklusive moms. Detta måste också betalas annars kan inte användarna skicka svar till textLocal när de vill acceptera en inloggning. [13]

4.3.3 SmartSMS

SmartSMS är ett svenskt företag som bedriver en tjänst för att skicka och ta emot SMS. Företaget har delat in sändning av SMS i olika prisklasser som de kallar kanaler. Dessa kanaler levererar olika prestanda gällande sändning av SMS. I den vanligaste kanalen kostar ett SMS inom Sverige 0,66 kr att skicka, och då kan man få statusuppdatering om SMSet har levererats eller inte. SmartSMS har även möjlighet till kostnadsfri SSL uppkoppling mot deras server. Vad gäller priser att ta emot SMS är detta inget som framgår av hemsidan.[14]

4.4 Valet av leverantör

Den leverantör som valdes efter testning var textLocal på grund av dess låga inträdelsekostnad, lågt pris inom England samt lättåtkomligt API. TextLocal har även haft bättre information på hemsidan vilket har gjort att man i ett tidigare skede har haft möjligheten att se hur man kan bygga upp systemet. Det har påvisats under utvecklingen att textLocal har mycket bra support som ofta ger svar på frågor inom två timmar.

4.4.1 Kostnad

Kostnaden för att textLocal skall stå för sändning samt mottagning av SMS är ungerfär 0.54 kronor per SMS, samt 398 kr i månaden exklusive moms för att kunna ta emot sms. Det finns billigare alternativ för att ta emot sms men då

brister en del av säkerheten.I de andra alternativen skickas nämligen sms som skall till olika servrar till ett och samma telefonnummer. Man har då en text i början av meddelandena i SMSen som specificerar en viss server. Detta gör att telefonnumren som SMSen skickas till används av flera servrar och man minskar antalet tecken som krypteringen kan bestå av eftersom både krypteringen och denna text skall dela på 160 tecken. Med det valda alternativet för att ta emot SMS hos textLocal har man sammankopplat ett visst telefonnummer med en viss server.

4.5 Krypteringen

Det finns tjänster som krypterar SMS-meddelanden för Androidtelefoner. Ett exempel är WISeSMS som använder sig av en AES(Advanced Encryption Standard) algoritm för att kryptera meddelanden. En AES kryptering går ut på att man ändrar klartexten genom en serie omväxlande substitutioner och permutationer [15] . En sådan serie kallas för en runda och det är olika antal rundor för olika AES-krypteringar. Ett annat exempel på tjänster som krypterar SMS är K-SMS[16] som är en betalversion men de beskriver inte vilken typ av kryptering de använder.

4.5.1 MCrypt

MCrypt är hämtat från androidsnippets.com vilket är ett forum där androidutvecklare delar kod och idéer tillsammans. MCrypt är en symmetrisk form av kryptering som bygger på AES-128. AES-128 betyder permutationer. AES-128 gör 10 stycken rundor och 128 står för antalet bitar nycklen till krypteringen består utav. 128 bitar översatt i tecken är 16 eftersom varje tecken är 8 bitar →16*8=128.
Fördelen med klassen MCrypt är att den har en klass för PHP och en klass för Java. Allting är open source och det finns hjälp att få hjälp på forumet angående algoritmen.[17]

4.5.2 Cryptastic

Andrew Johnson har använt Mcrtpy som är ett standardbibliotek inom PHP. Det är alltså inte den klass som är beskrivet i 3.5.1 utan en standardklass som finns i PHP med samma namn. Han har gjort en klass kallad cryptastic med kryptering/dekryptering för PHP. Cryptastic har tre olika funktioner. En funktion som genererar en nyckel som används när man skall kryptera och dekryptera meddelanden. Funktionen har två texter som inparametrar som används för att generera nyckeln. Eftersom krypteringens metod är symmetrisk kommer både mobiltelefonen samt servern innehålla samma nyckel och därav samma parametrar för generering av nyckeln. De andra två funktionerna är kryptering respektive dekryptering. Som indata har de nyckeln samt den text som skall krypteras/dekrypteras.

Fördelen med denna form av kryptering är att den är lätt att använda samt är helt open source. Koden är väldokumenterad och det är lätt att sätta sig in i programmet.[18]

4.6 Testning av krypteringen

De olika krypteringsmetoderna testades i PHP via en egentillverkad hemsida i PHP som visar klartext och kryptering. Det som främst testas är om det bildas onödigt många tecken från en given text. Det främsta målet är att SMSet inte ska bestå av fler tecken än 160. Om ett SMS skulle överskrida 160 tecken skulle varje SMS som skickas till eller från servern vara dubbelt så dyra, vilket gör tjänsten dubbelt så dyr. Med PHP-sidan kan man även se vilka tecken den krypterade texten består av. Om det bildas tecken utöver standarden GSM 7-bit Alphabet så kommer SMSet att gå över till unicode SMS.

4.7 Testresultat

Vid testning av de olika krypteringarna visade det sig att MCrypt var det bästa alternativet. En av anledningarna att MCrypt valdes var att koden var i stort sätt färdig vid implementation för både java och PHP. Cryptastic var endast utvecklad för PHP vilket hade gjort att mycket jobb var kvar för att javaprogrammet i mobiltelefonen skulle kunna kommunicera med servern. Det visade sig att texten endast genererade värden i hexadecimal formvilket lämpar sig bra inom GSM 7-bit alfabetet. Detta medför att man kan använda 160 tecken för att kryptera texten. Koden består av endast öppen källkod vilket gör att den tillåter modifieringar.

5 Implementering

Kapitlet beskriver hur systemet implementerades samt förklarar varför det implementerades på detta sätt.

5.1 Androidapplikationen

5.1.1 Krav och mål

Det fanns inga tydliga krav på Androidapplikationen innan den skulle utvecklas. Däremot fanns det mål som visade hur en idealisk applikation skulle se ut efter utvecklingen. Man ville att applikationen skulle automatiskt ställa en förfrågan till användaren när ett godkänt SMS kom från servern eftersom det är väldigt bekvämt för användaren. Man ville även att SMSen inte skulle synas när de anlände till telefonen och inte heller synas bland användarens inkomna meddelanden eftersom man då kan se vart SMSen skickas samt krypteringarna som skickas till och från servern.

5.1.2 Utvecklingen

Utvecklingen skedde i Eclipse och simulerades i virtuella enheter. Virtuella enheter är virtuella mobiltelefoner och ingår i den utvecklingsmiljö som man laddar ner när man skall utveckla program för Android. Eftersom författaren inte hade jobbat med Androidutveckling tidigare så bedrevs utvecklingen parallellt med att författaren gjorde övningsuppgifter och följde forum på internet. Det fanns även personer att fråga på Crunchfish i Malmö som har stor erfarenhet av Androidutveckling.

5.1.3 Resultat

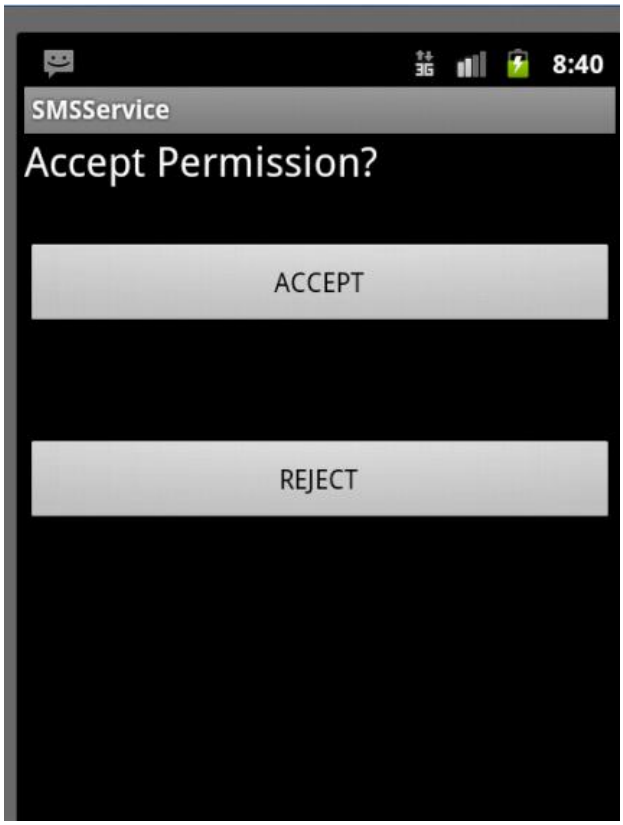
Applikationen blev som förväntat där vissa av de mål som sattes upp klarades av. Här nedan följer en enkel beskrivning av applikationen.

5.1.4 Applikationen

Applikationen installeras på mobiltelefonen. Om användaren försöker starta applikationen från användarens meny kommer endast ett fönster fram. Fönstret innehåller en text som säger att användaren måste göra ett inloggningsförsök på en hemsida för att man skall kunna logga in med applikationen.

Figur 10. En splashscreen som visas när en användare startar applikationen.

När en användare har skrivit in sitt användarnamn på en kunds hemsida, skickas ett SMS från textLocal till användarens mobiltelefon. Mobiltelefonen granskar alla inkomna SMS om det är rätt telefonnummer och om dekrypteringen av meddelandetextexten ger det förväntade resultatet.
Om detta stämmer kommer en ruta upp som frågar användaren om han eller hon vill acceptera uppkopplingen. Om användaren valde att acceptera uppkopplingen till hemsidan så skickas ett krypterat SMS tillbaka till servern som säger att man tillåter uppkopplingen. Om användaren nekar inloggningen till hemsidan skickas inget SMS till servern.



Figur 11. En layout som kräver en bekräftelse från användaren.

Det som inte implementerades i applikationen var funktionalitet för att SMSen skall vara dolda för användaren. Detta gjordes inte eftersom man inte hittade en bra lösning för att dölja SMSen.

5.2 PHPservern

De funktioner som implementerades i PHP var följande:

5.2.1 SendSMS

SendSMS funktionen fungerar nästan på samma sätt som när man skall skicka ett vanligt SMS från en mobiltelefon. Som inparametrar har den ett telefonnummer vilket är det telefonnummer som man skall skicka ett meddelande till och en text vilket är den hemlighet som finns i mobiltelefonen. Denna text krypteras och skickas tillsamans med telefonnummret till textLocal.

5.2.2 ReciveSMS

ReciveSMS har fyra in-parametrar. Den första parametern är den hemlighet som skickades. Den andra är den inkommande krypterade text som skall vara svar på den skickade hemligheten. De andra två parametrarna är det telefonnummer som hemligheten skickades till och den andra är det telefonnummer som den inkomna krypterade texten kom ifrån. Sedan kollar funktionen om telefonnummren stämmer samt dekrypterar det inkomna meddelandet och granskar ifall det är rätt svar på den angivna hemligheten.

5.2.3 SmsAuthentication

SmsAuthentication är den funktion som sköter kallelse mellan reciveSMS och sendSMS. Detta är en funktion som måste ändras mycket när systemet får flera användare.

5.3 Vidarebefodring från textLocal till Ubiqos server

Denna del av implementeringen blev aldrig färdig. Detta berodde på att man hade problem med den testserver som användes. Först användes en WAMP server som den största delen av utvecklingen skedde på. Men när denna server skulle läggas upp online upstod komplikationer och därför lyckades aldrig servern läggas upp på internet. Detta kan bero på att en annan icke fungerande server hade använts på den router som användes för att koppla upp WAMP-servern. Det är möjligt att denna router innehåller inställningar som förhindrar WAMP-servern att läggas upp på internet. Trots att inställningarna kontrollerades flera gånger kunde problemet inte lösas.
För att lösa detta köptes en domän på one.com. Detta gav tyvärr inte heller något bra resultat. Efter diskussion med one.coms support framgick att det var möjligt att någon av one.coms brandväggar blockerade denna inloggning. Anledningen till att brandväggar ofta blockerar sådana anslutningar är att servrar får tillåtelse att skriva rakt in i det program som skall använda informationen.
De implementeringar som skall ändras för att vidarebefordringen skall fungera är följande.

1. En fil med namnet script.php skall ta emot innehållet som kommer från textLocal.
2. Filen skall innehålla följande kod:

```
<?php
$sender = $_REQUEST['sender'];
$content = $_REQUEST['content'];
$inNumber = $_REQUEST['inNumber'];
$email = $_REQUEST['email'];
$credits = $_REQUEST['credits'];
?>
```

Figur 12. Kod tagen från textLocals API http://www.textlocal.com/developers/code/

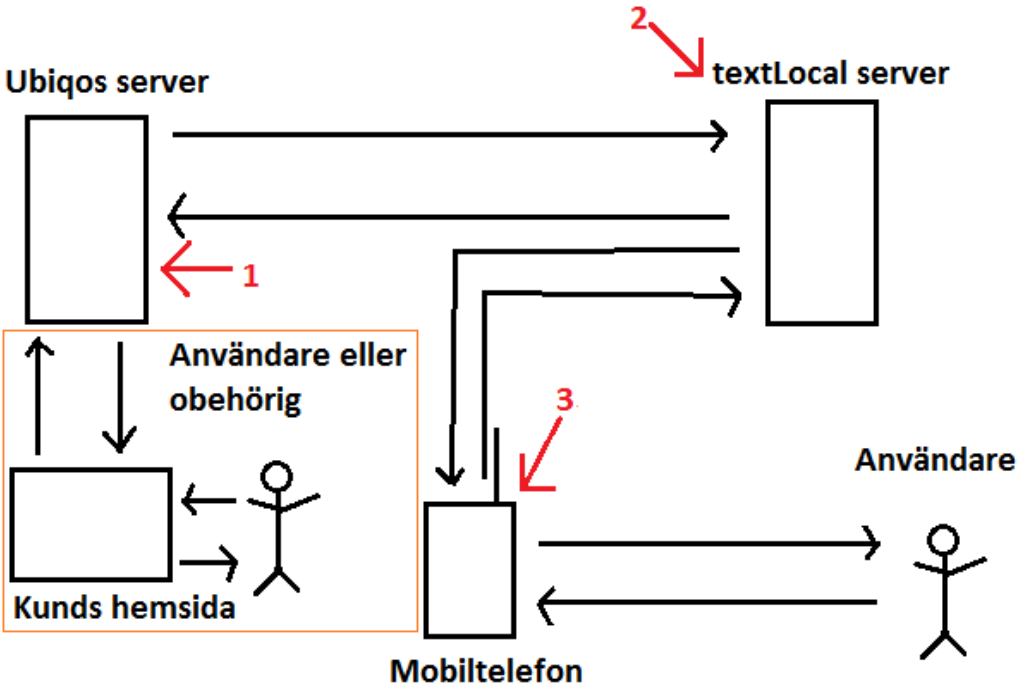
Koden sparar undan information från textLocal i variabler.

3. Hemsidans URL skall skrivas in på textLocals hemsida. Först måste man logga in och sedan navigera fram till fliken inställningar på inkorgen.
4. Sedan skall $sender och $content skickas vidare till reciveSMS funktionen i filen smsAuthentication.php.
5. Det skall även implementeras en sats som verifierar att informationen kommer från textLocal. Denna information går att nå genom att kontakta textLocals support.

6 Säkerhetsgranskning av systemet

Detta kapitel analyserar säkerheten i systemet. Den visar möjliga sätt att knäcka systemet samt motåtgärder till detta. Viktigt att påpeka är att detta är en bild av systemet när det är helt färdigutvecklat. Vi antar här att man har gjort systemet kompatibelt att hantera flera mobiltelefoner.

Man kan säga att information som går till mobiltelefonen är i allmänhet ointressant eftersom det endast har som uppgift att kräva ett svar från telefonen. Enda möjligheten för den informationen att vara skadlig är att manipulera applikationen genom att knäcka krypteringen och på så sätt skicka in skadlig programkod via SMEen. Men när väl informationen nått mobiltelefonen har den fått nyckelvärden. Det är dessa nyckelvärden som ger en specifik användare tillträde till hemsidan.



Figur 12. Visar eventuella ingångar för att hacka systemet.

1. Ett effektivt sätt att knäcka systemet är att direkt kommunicera med Ubiqos server. På detta sätt kan man försöka mata in rätt information efter man har begärt att logga in på en hemsida. Motåtgärder till detta är att man kräver att förbindelsen kommer från textLocals server samt att det är rätt telefonnummer och hemlighet i det krypterade meddelandet. Detta krypterade meddelandet är unikt för varje mobiltelefon vilket gör att om man lyckas få reda på en användares hemlighet kan inte detta användas för att logga in på någon annans hemsida.

2. Man kan knäcka lösenordet och användarnamnet till textLocal. Det finns känslig information hos textLocal angående telefonnummer och krypteringar som passerar genom tjänsten. Om man läser av inkomna SMS från en given mobiltelefon så kan man knäcka krypteringen. Man kan också byta användarnamn och lösenord på textLocal vilket gör att tjänsten blir obrukbar. Om detta inträffar måste man skapa ett nytt konto med ett nytt användarnamn och lösenord hos textLocal. Detta motverkas genom att välja ett så svårknäckt användarnamn och lösenord som möjligt. Detta användarnamn och lösenord behöver endast servern veta om och den person som står för driften av systemet.

3. Man kan försöka knäcka systemet genom att bygga en egen applikation som kan läsa av de inkommande SMSen automatiskt. Man måste då ha ett godkänt konto registrerat som en användare av systemet. Om man inte har det måste man fiska efter SMS som skall komma till en specifik användare vilket är väldigt tidskrävande. Anledningen till att detta är tidskrävande är att det är massvis med SMS som skickas regelbundet både in och ut från textLocals servrar. Att fiska rätt på SMS som skall användas av Ubiqos tjänst blir därför väldigt problematiskt och än värre att hitta information som skall till en given telefon. Men har man ett godkänt konto från ett företag kommer den angivna telefonen kommunicera med Ubiqos samt textLocals server. Användaren kan då lyckas knäcka krypteringen. Om man knäcker krypteringen för sin mobilapplikation kan man använda denna lösning för att knäcka en annan användares kryptering.

7 Vidareutveckling av systemet

Systemet är nu gjort i ett testscenario. Denna version går bra att installera hos en kund, men framförallt att använda som visning i framtida planer för Ubiqos investerare och sammarbetspartners. Detta kapitel innehåller information om delar som måste förbättras på systemet samt visioner på ett framtida system.

7.1 Säkerhetsaspekter

Rent säkerhetsmässigt är det ett antal punkter som måste förbättras för att få fram ett säkrare system.

7.1.1 Krypteringen

Krypteringen som används inom systemet idag är en kryptering som finns tillgänglig på internet. Det vore bra ifall man försöker knäcka denna kryptering för att garantera att den är tillräckligt säker. Krypteringen borde även innehålla information om vilken sida det är som vill ha bekräftelse på inloggning. Detta kan senare visas för användaren vid bekräfteslen av uppkopplingen, exempelvis ”Grant permission to Volvo?”. Användaren kan då se i mobiltelefonen om det verkligen är Volvo som vill komma åt inloggning och inte något annat företag.

7.1.2 TextLocal

TextLocal används idag som leverantör av GSM kommunikationen mellan mobiltelefon och server. TextLocal har en vanligt inloggningstjänst med användarnamn och lösenord för att autentisera en användare. Om man lyckas knäcka detta användarnamn och lösenord erhålls kontroll över TextLocal-kontot och därav möjlighet att stänga av funktionen som att skicka SMS. Den optimala lösnigen på problemet hade varit att själv stå för GSM-kommunikationen. Men detta hade ökat kostnaderna för systemet avsevärt. Ett bra alternativ hade varit att kontakta exempelvis textLocal och fråga ifall man kan öka säkerheten vid autentisering mot dem på något lämpligt sätt. Användarnamn och lösenord kan ge en bra form av säkerhet om man exempelvis kan ha väldigt stora lösenord med olika tecken.

7.1.3 Unika applikationer

I AES-krypteringen finns det en secret key som är likadan i applikationen och PHP-servern. Det är denna secret key som gör att krypteringen är unik. Om man lägger till ytterligare användare i systemet kommer alla telefoner ha samma kryptering med servern. Detta resulterar i att om man har knäckt krypteringen för en användares telefon har man också knäckt alla användares kryptering mot servern. Man skulle behöva en funktion som genererar unika applikationer automatiskt, där inparameter för funktionen skall vara användarens telefonnummer. Det funktionen skall generera är en hemlighet och secret key för varje mobiltelefon. Denna Secret key måste hårdkodas i

applikationen men kan finnas som inparameter när man kör Crypt klassen på PHP-servern. Hemligheten skall vara inparameter till funktionen sendSMS som är gjord i detta examensarbete. Detta resulterar i att både kryptering och hemlighet kommer vara unika.

7.2 Ekonomiska aspekter

7.2.1 Sändning av SMS

I dagsläget skickas SMS från England, oavsett vart någonstans i världen användaren befinner sig. Detta betyder att det tillkommer extra taxa när man skickar ett SMS från England till Sverige och tvärt om. Det vore bättre att ha en inbyggd funktion som märker vilken land användarens telefonnummer tillhör och senare skickar ett SMS från en SMS-leverantör från det landet.

7.3 Funktionella aspekter

Funktionella aspekter innefattar funktioner som måste implementeras för att prototypens funktionalitet skall vara av högre kvalité. Funktionerna är indelade i kritiska aspekter vilka måste implementeras innan man kan börja lansera tjänsten samt icke-kritiska aspekter som kan implementeras vid ett senare tillfälle i form av uppdateringar.

7.3.1 Kristiska aspekter

Kritiska aspekter inefattar både implementation, samt mer systemegenskaper om hur systemet skall fungera och kallas. Dessa punkter måste implementeras innan man kan börja lansera systemet.

7.3.1.1 Implementering

Man måste implementera hur och när tjänsten skall kallas. Skall användaren själv tala om för systemet vid identifiering (inmatning av användarnamn på kundens hemsida) när SMS skall skickas eller märker systemet själv när användaren inte svarar på sin mobiltelefon att ett SMS skall skickas? Skall tjänsten vara tillgänglig för alla som använder Ubiqos system, eller skall det vara tillgängligt för endast vissa användare som betalar en summa för att få använda GSM-funktionen?

7.3.1.2 Användargränssnittet

Det är viktigt att användaren har möjlighet att stoppa inloggningen mot hemsidan om han/hon inte vill logga in medans man väntar på att få ett svar från användarens mobiltelefon. GSM-nätet kan vara överbelastat vilket gör att det tar lång tid innan man lyckas skicka SMS till och/eller från mobiltelefonen. Om en användare märker att det tar lång tid att logga in borde användaren kunna avbryta inloggningen. Det borde även finnas en timer som stänger av inloggningen efter exempelvis 4 minuter. Det finns annars en risk att en användare lämnar datorn under en inloggning och att ett SMS erhålls senare. Om en obehörig då passerar datorn kan han/hon logga in på tjänsten.

7.3.1.3 Spärrning av versioner

Det borde implementeras en funktion som gör en version av applikationen otillgänglig om säkerhetsbrister påträffas i den versionen. Visuellt behövs en ruta som kommer fram vid försök av autentisering som säger att en ny version av applikationen behövs laddas ner.

7.3.2 Icke kritiska aspekter

De icke kritiska aspekterna kan implementeras när antalet användare för systemet växer.

7.3.2.1 Användargränssnittet

Man hade kunnat integrera bilder i samband med pushnotification fönstret som begär inloggning. Exempelvis om man försöker logga in på Volvo så kan Volvos logga komma upp. Denna logga skall då lagras i telefonen och lämpligen nås genom att länka samman företagsnamnet med en logga i en mindre databas. Om en implementering av detta sker borde man ha uppdateringar i åtanke eftersom det förmodligen anses störande om applikationen begär uppdateringar varje gång en ny logga skall läggas in i applikationen.

8 Resultat

Examensarbetets resultat blev en nästintill färdig prototyp som kan användas vid autentisering. Prototypen bygger på att man har en mellanhand mellan den server och mobiltelefon som vill autentisera varandra. Servern och mellanhanden kommunicerar med varandra via en krypterad internetuppkoppling och mellanhanden och mobiltelefonen kommunicerar via GSM-nätet. De huvudsakliga beståndsdelar som gör prorotypen säker är följande.

1. Servrarna kommunicerar med varandra via en SSL-krypterad anslutning.
2. Delar av den information som skickas mellan server och mobiltelefon har minst en kryptering som skyddar viktiga delar av informationen.
3. Man kräver att all information inom systemet kommer från rätt avsändare.

Det var en förbindelse servrarna emellan som inte implementerades i systemet på grund konfiguration av brandväggar och routrar.
Examensarbetet innehåller information och ideér angående utveckling av systemet. Denna information är intressant att ta del utav och vissa punkter måste implementeras för att systemet skall vara praktiskt dugligt. En stor fördel med systemet är att det är väldigt anpassningsbart och flexibelt samt har en stor kapacitet ifall trafiken skulle öka drastiskt. En nackdel är dock att systemet kostar 480 kronor i månaden samt ytterligare 0.65 kronor per SMS som skickas till Sverige. Det är även en kostnad för den användare som väljer att använda tjänsten. Denna kostnad är beroende på vad användaren har för operatör och abonnemang men kan upskattas till 0.70 kronor per inloggning.

9 Slutsats

Det är möjligt att utöka Ubiqos autentiseringstjänst med hjälp av GSM. Den prototyp som gjordes i detta examensarbete förlitar sig på en mellanhand som möjliggör kommunikation mellan internet och GSM. Lösningen som är gjord kostar 480 kronor i månaden samt ytterligare 0.65 kronor per SMS som ska till Sverige. Det tillkommer även en kostnad för användare som vill logga in med tjänsten som är varierande men borde ligga mellan 0.5-0.9 kronor/ autentisering. Delar som behöver justeras för fullskalig implementering samt idéer som ökar säkerhet och funktionalitet för den framtida produkten finns dokumenterat i denna rapport. Förhoppningen är att Ubiqo skall få ett positivt gensvar hos kunder och investerare vilket leder till att man vill möjligöra autentisering via GSM. Om en implementering av tjänsten skall genomföras finns mycket information att tillgå från denna rapport. Man kan också tänka sig att en annan högskoleingenjör hade kunnat vidareutveckla tjänsten i ett annat examensarbete för att färdigställa tjänsten.

10 Terminologi

1. LAMP = Linux Apache MySQL PHP/Python/Pearl. LAMP är en vanlig form av mjukvara för servrar.
2. WAMP = Windows Apache MySQL PHP/Python/Pearl. Wamp är en vanlig form av mjukvara för servrar.
3. SMS = Short Messaging Service. En tjänst som finns inom GSM. Den ger möjlighet att skicka små textfiler på 160 tecken inom GSM.
4. MMS = Multimedia Messaging Service. MMS och fungerar som SMS med skillnad att den kan skicka större mängd data. Detta ger möjlighet att skicka bilder samt mindre videofilmer.
5. PHP = Hypertext preprocessor. Är ett skriptspråk som är vanligt inom webbprogrammering.
6. AES = Advanced Encryption Standard. En krypteringsmetod som använder ett symmetrikst blockkrypto och finns i versioner med olika storlek på nycklarna för kryptering/dekryptering.
7. CEPT = Conférence Européene des Postes et Télécommunications. En organisation som har funnits sedan 1959. Den består av flera olika länder som diskuterar telefonkommunikation inom Europa.
8. Kanban = En projektmodell som använts flitigt av Toyota. Man kan sammanfatta den med en projekttavla där man kan se vad som produceras samt vad som måste produceras.
9. Scrum = Scrum är en iterativ projektmodell som bygger på att man delar upp projekt i olika delar som man jobbar med. Den innehåller regler som hjälper till så att utvecklingen går frammåt.
10. TDMA = Time Division Multiple Access. TDMA är ett sätt att sända olika signaler på en och samma uppkoppling.
11. Kodex = En slags motor som behövs för att läsa av viss musik och videofilmer.
12. Ordlista attack = Ett sätt som används för att knäcka lösenord. Man testar att mata in ord som finns in en specifik lista.
13. Brute force attack = Ett sätt som används för att knäcka lösenord.Man matar systematiskt in olika kombinationer av bokstäver för att tillslut knäcka lösenord…
14. Symmetrisk form av kryptering = En form av kryptering där båda parterna som kommunicerar med varandra använder samma nycklar för att kryptera och dekryptera meddelanden.
15. Hexidecimalavärden = Tal som har 16 som bas än de normala talen med 10 som bas. Värdena består av 0 till 9 samt a till f.
16. Klartext = En klartext är en text som går att läsa utan problem. En icke krypterad text.

11 Referenser

[1] Crunchfish hemsida.
http://www.crunchfish.com/
2012-06-04
[2] Ubiqos hemsida.
http://www.ubiqo.se/
2012-06-04
[3] Ett examensarbete inom samma område.
http://homepages.mcs.vuw.ac.nz/~ian/shared/papers/secureweb.pdf
2012-06-04
[4] Ett examensarbete inom samma område.
http://people.scs.carleton.ca/~mmannan/mpauth/mpauth-extended.pdf
2012-06-04
[5] Information angående GSM.
Student Text GSM system introduction: Ericsson Radio Systems AB: 1998 :
EN/LZT 123 3641 R2B: sida 6
http://www.cept.org/cept/about-cept
2012-06-04
http://www.webcitation.org/5yRQRGPgH
2012-06-04
[6] Information om A5/1-kryptering.
http://cryptome.org/a51-bsw.htm
2012-06-04
[7] Beskrivning hur Karsten Nohl knäckte A5/1 kryptering.
http://srlabs.de/blog/wp-content/uploads/2010/07/Attacking.Phone_.Privacy_Karsten.Nohl_.pdf
2012-06-04
[8] Information angående A5/2 kryptering.
http://cryptodox.com/A5/2
2012-06-04
[9] Ett företag som gör mjukvara vid exempelvis sändning av SMS via GSM och GPRS-modem.
http://www.activexperts.com/mmserver/
2012-06-04
[10] Information om GPRS.
http://www.telecomspace.com/datatech-gprs.html
[11] Källor som användes vid PHP programmering.
http://se2.php.net/manual/en/intro-whatis.php
2012-06-04
http://se2.php.net/manual/en/intro-whatcando.php
2012-06-04
Webbprogrammering med PHP av Morgan Augustsson och

Stefan Folkesson ISBN10:9163609738 utgiven 2010-11
[12] Information om Androidprogrammering.
http://developer.android.com
2012-06-04
http://developer.android.com/guide/basics/what-is-android.html
2012-06-04
[13] TextLocals hemsida.
http://www.textlocal.com/
2012-06-04
[14] SmartSMSs hemsida.
http://www.smartsms.se/
2012-06-04
[15] Information om wiseSMS.
http://www.androidzoom.com/android_applications/communication/wisesms-lite_badnw.html
2012-06-04
[16] Information om KSMS.
http://www.androidzoom.com/android_applications/communication/ksms_booel.html
2012-06-04
[17] Klassen MCrypt
http://www.androidsnippets.com/encrypt-decrypt-between-android-and-php
2012-06-04
[18] Klassen Cryptastic.
http://www.itnewb.com/tutorial/PHP-Encryption-Decryption-Using-the-MCrypt-Library-libmcrypt
2012-06-04
[19] Information om LAMP-server
http://www.lamphowto.com
2012-06-04