УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП
ФАКУЛТЕТ ЗА ИНФОРМАТИКА
КАТЕДРА ПО ИНФОРМАЦИСКИ СИСТЕМИ
ШТИП

Сашо Ѓеоргиевски

ФОРЕНЗИЧКИ МЕТОДИ ЗА АНАЛИЗА НА УРЕДИ СО iOS ОПЕРАТИВЕН СИСТЕМ

МАГИСТЕРСКИ ТРУД

Штип, Јуни 2015 година

САШО ЃЕОРГИЕВСКИ

ФОРЕНЗИЧКИ МЕТОДИ ЗА АНАЛИЗА НА УРЕДИ СО iOS ОПЕРАТИВЕН СИСТЕМ

УНИВЕРЗИТЕТ „ГОЦЕ ДЕЛЧЕВ“ – ШТИП

Комисија за оценка и одбрана

Ментор:	Д-р Сашо Коцески Вон.Проф., Универзитет ,,Гоце Делчев" - Штип, Факултет за Информатика
Член	Д-р Александра Милева Вон. Проф., Универзитет ,,Гоце Делчев" - Штип, Факултет за Информатика
Член	Д-р Југослав Ачковски Доцент, Воена академија „Генерал Михајло Апостолски"- Скопје

Членови на комисија за оценка и одбрана

Претседател:	Д-р Александра Милева Вон.Проф., Универзитет ,,Гоце Делчев" - Штип, Факултет за Информатика
Член	Д-р Сашо Коцески Вон.Проф., Универзитет ,,Гоце Делчев" - Штип, Факултет за Информатика
Член	Д-р Југослав Ачковски Доцент, Воена академија „Генерал Михајло Апостолски"- Скопје
Научно поле:	Компјутерска техника и информатика
Научна област:	Информатика, Процесирање на податоци, Информациони системи и мрежи
Датум на одбрана:	јуни, 2015 година
Датум на промоција:	јуни, 2015 година

Изразувам голема благодарност кон менторот Вон. Проф. Д-р Сашо Коцески за неговото несебично залагање и ангажирање за време на изработката на овој магистерски труд. Благодарност до членовите на комисијата чии предлози, стручни мислења и забелешки ми помогнаа во текот на подобрување на магистерскиот труд. Благодарност кон моето семејство и најблиските, особено кон мојата сопруга Марија и ќерка Неда, кои безрезервно ме поддржуваа и разбираа при изработката на овој магистерски труд.

Содржина

КРАТОК ИЗВАДОК

Употребата на дигиталните уреди сè повеќе се зголемува како што луѓето во својот живот ја интегрираат технологијата. Со развојот на новите технологии, криминалците наоѓаат начин да ги искористат технологиите за извршување на криминалните дејства. Барањата за мобилна форензика драстично се зголемија со излегувањето на паметните телефони поради напредните компјутерски способности, големиот простор за складирање податоци, интерфејс на допир. Поради сите овие карактеристични способности, iPhone е меѓу најмногу користените паметни телефонски уреди. Целата комуникација на овие уреди е документирана, бидејќи телефоните повеќе не се користат само за зборување. Секогаш кога ќе се оствари телефонски повик, ќе се испратат текстуални пораки, проверка на и-мејл, пребарување низ интернет, не размислуваме дека сите овие податоци се зачувуваат на самиот уред. Доколку овие податоци се избришат, тие можат да бидат вратени во целост. Со овој документ за форензичка анализа на IOS ќе бидат опишани сите функционалности на самиот уред кој претставува предмет на форензичка анализа, со цел да се сфати начинот на работа за да можеме на соодветен начин и со претходно дефинирани методи да пристапиме кон негова анализа.

КЛУЧНИ ЗБОРОВИ

iPhone, iDevice, Apple, forensics, image, keychain, HFS Plus, Journaling, Firmware, Jailbroken, SQLite

ABSTRACT

The use of digital devices is increasing as people in their lives are integrating the technology. With the development of new technologies, criminals find ways to use technology to carry out criminal activities. Requests for mobile forensics soared with the release of smart phones with advanced computing capabilities, a large space for data storage, the touch interface. For all these distinctive capabilities, iPhone is among the most used smart phone devices. All communication on these devices has been documented since the phones are no longer used just for talking. Whenever you have a phone call, sending text messages, checking e-mail, searching the Internet, we are not aware that all these data are being saved on the device. If these data are deleted, they can be recovered. This document for forensic analysis of iOS will describe all features of the device that is the subject of forensic analysis in order to understand how we can work properly with predefined methods to approach to its analysis.

KEYWORDS

iPhone, iDevice, apple, forensics, image, keychain, HFS Plus, Journaling, Firmware, Jailbroken, SQLite

ВОВЕД

Мобилните уреди во изминативе неколку години се користеа само за телефонски разговори, но како што се усовршуваа нивните можности за праќање и примање текстуални пораки, зачувување контакти, фотографирање, барање одредена локација на мапите, статуси на Facebook итн., ги правеа мобилните уреди лесно достапни. Apple има создадено прекрасни уреди, кои беа замислени да им помогнат на многу луѓе. Производите на Apple, како што се iPhone и iPad, се многу корисни алатки и во суштина се преносни компјутери на кои работи послаба верзија на UNIX и MAC OS X оперативниот систем. Мобилните уреди на Apple со можност за мрежно поврзување, со внатрешна флеш меморија од 8GB до 64GB, со GPS функционалност и вградена камера, содржат многу податоци, зачувани при секое користење на нивните функционалности. Дел од овие податоци се телефонски повици, контакти, текстуални пораки, геолокациски координати, фотографии и друго. Како и повеќето оперативни системи, бришењето на датотеките е бришење на референцата до тие податоци. Поради ова работат програмите за обновување на податоците. На iPhone, количината на податоци е многу поголема од податоците што се достапни преку корисничкиот интерфејс. iPhone и iPad многу бргу станаа лидери на пазарот за мобилна технологија, со широк опсег на функционалности, а комбинирани со нивната мобилност се користат како мобилна канцеларија. Цената на оваа продуктивност е зачувувањето на осетливите информации на овие уреди. Со сите убави и прекрасни работи, постојат поединци кои овие пронајдоци на денешницата ги користат за остварување криминални цели. Овие уреди се општествена појава и се растечкиот дел од мобилните телефони и во последно време сè повеќе ги гледаме како одземени уреди за вештачење во криминалистичките лаборатории. Употребата на дигиталната форензика е ефективна алатка во спроведување истраги и проценки за претходни преземени активности.

ЦЕЛ НА ИСТРАЖУВАЊЕТО

Мобилните уреди, особено паметните телефони со функции како една мини канцеларија, многу брзо стануваат секаде присутни кај популацијата низ целиот свет. Овој успех на мобилните телефони се должи на фактот дека телефонот помага во задоволување на човечките потреби, бидејќи денес мобилните телефони се сè помалку телефони, а сè повеќе се преносни компјутери. Ова ги прави мобилните уреди потенцијален извор на вредни докази. Apple преносните уреди ги има во различни големини со различни карактеристики, но заеднички им е оперативниот систем iOS, кој е достапен само за уредите на Apple, за разлика од оперативниот систем Android, развиен од Google, кој се користи на различни хардверски платформи произведени од најразлични фирми.

Во последниве неколку години, забележуваме значителен раст на побарувања за извлекување податоци од iPhone уредите. Со овој раст се појавува и потребата за развој на процес и насоки за испитување на овие уреди. Додека кај компјутерската форензика се извлекува тврдиот диск, се приклучува на уред наречен writeblocker[1] и со користење специјален лиценциран форензички софтвер, се врши физичко или бит по бит копирање на дискот и хеширање, како доказ дека копираната содржина е идентична со содржината од дискот кој се копира, кај мобилната форензика често има потреба уредот да се исклучи и повторно да се вклучи со што се врши промена на податоците во самиот уред. Часовникот на мобилниот уред постојано се менува и со тоа се прави ажурирање на податоците на тој уред, па поради тоа форензичката хеш вредност ќе биде различна секојпат кога ќе се примени истата хеш функција [11]. Во мобилната форензика, особено кај iPhone уредите каде сите податоци се шифрирани и се недостапни за нивно извлекување, развојот на процес и насоки за форензика на iPhone уредите е голем предизвик.

[1] Форензички уред на кој се прикачува уредот од кој треба да се извлечат податоците. Намената на овој уред е тоа што не дозволува да се направи запис врз уредот од кој се копираат сите податоци.

1. ДИГИТАЛНА ФОРЕНЗИКА

Брзиот развој на технологиите е воден од желбата за мала потрошувачка, голем капацитет и голема ефикасност. Оваа рапидна еволуција на компјутерите и мобилните преносни уреди е причината за нивното користење во криминални активности. Употребата на дигиталните уреди се зголемува со интегрирањето на технологијата во секојдневниот живот. Зголеменото барање за вештачење на дигитални уреди е доказ за големата распространетост на дигиталните уреди во општеството. Дигиталната форензика е [3] гранка на форензичката наука која вклучува прибирање, складирање, анализирање и презентирање на доказите од дигиталните медиуми. Во истрагите на модерниот криминал, дигиталната форензика има голема улога, бидејќи луѓето сè повеќе ги користат функционалностите на дигиталните компјутерски уреди. Овој термин се користеше како синоним за компјутерската форензика, но со проширувањето ги опфаќа сите уреди со можност за зачувување дигитални податоци како: компјутери, принтери, мрежни уреди, мобилни телефони, дигитални преносни уреди, GPS уреди и др. Дигиталната форензика постои скоро од измислувањето на компјутерите. Но, стануваме сведоци на надмоќта врз дигиталната форензика и форензичките способности поради брзиот развој на технологиите. Дигиталната форензика е апликација од научни методи за процесот на извлекување податоци од дигиталните уреди. Фокусот на компјутерската форензика се точно одредени методи на извлекување податоци од точно одредена платформа, додека дигиталната форензика ги опфаќа сите дигитални уреди без дефинирани конкретни методи, процедури или алатки. Потребно е да се установат стандарди или методи, како дигиталните податоци ќе бидат документирани, извлечени и зачувани со соодветни специјални алатки, а притоа да не биде компромитиран интегритетот на доказите. Еден од принципите на дигиталната форензика е да се гарантира дека оригиналниот медиум нема да биде променет и методите користени за креирање форензичка копија од медиумот за интегритетот на

оригиналот. Процесот на дигиталната форензика се состои од следниве чекори [4]:

- Запленување или одземање – процес на стопирање активности кои можат ги оштетат дигиталните информации. Се врши одземање на предметниот доказен предмет;
- Правење форензичка копија – бидејќи информациите се зачувани на дигитален медиум, тој ќе биде отстранет од запленетиот уред и со приклучување на writeblocker и со соодветна софтверска алатка ќе биде направена форензичка копија од дигиталниот медиум;
- Анализа на форензичката копија – систематско пребарување на доказите поврзани со случајот што се истражува;
- Извештај – за целиот истражувачки процес, како и најдените податоци со нивна точна локација на медиумот.

1.1. МОБИЛНА ФОРЕНЗИКА

Денешните мобилни уреди се софистицирани преносни уреди со можност за складирање голема количина на податоци. Сè повеќе се зголемува нивната побарувачка во однос на преносните компјутери, бидејќи главни одлики им се големата меморија, лесната преносливост, ниската цена и лесната достапност. Овие мобилни уреди се едни од главните комуникациски уреди во секојдневниот живот. Со секоја испратена или примена текстуална порака, пребарување на интернет и др., самите како корисници не сме свесни дека сите овие активности се зачувуваат во меморијата од уредот. Денеска сите ние на нашите мобилни уреди чуваме информации со детали за секој аспект од нашиот живот. Добивањето пристап до овие информации во текот на истрагата е многу важно за обезбедување докази за конкретниот случај. Достапноста на мобилните уреди во комбинација со напредните можности на овие уреди е причината за развој на форензиката на мобилните уреди. Форензиката на мобилните уреди [4] е област од дигиталната форензика која врши извлекување дигитални докази или податоци

од мобилните уреди во форензички услови. Поимот мобилни уреди се однесува на мобилните телефони, дигитални уреди со комуникациски способности, таблети, GPS уреди и др. За нас како форензичари, вештачењето на мобилните уреди во однос на компјутерите претставува голем предизвик со оглед на тоа што надградбата и усовршувањето на компјутерите се состои во зголемување на RAM меморијата или просторот на тврдиот диск, а со тоа компјутерската технологија останува иста. Кај мобилните уреди често се менува оперативниот систем, хардверот, мемориската технологија и интерфејсот за комуникација. Мобилните уреди се разликуваат и во дизајн, но, исто така, технологијата секојдневно се менува. Кај мобилните уреди најчесто користени оперативни системи се Apple iOS, Android, RIM, Symbian и Windows phone [12]. Класично кај компјутерските системи се отстранува дискот и се прави форензичка копија, но кај мобилните уреди често предметниот уред мора да се вклучи заради подигнување на оперативниот систем и да се работи врз него без writeblocker, а притоа внимавајќи на интегритетот на податоците. Форензиката на мобилните уреди наметнува комплетно различен форензички процес во однос на компјутерската форензика. Според National Institution for Standards and Technologies (NIST) [4], не постојат стандардни техники, методи или процедури кои би се користеле за сите мобилни уреди. Можеме да кажеме дека постојат различни методи и пристапи за уреди со ист оперативен систем поради разликите во самиот хардвер и дизајн. Поради ова, мора да разбереме како работат мобилните уреди и на кој начин ги складираат податоците со цел создавање форензички услови за пристап кон извлекување на истите податоци. Постојат различни методи за извлекување податоци од мобилните уреди, но најчесто извлечени податоци се телефонските контакти, текстуални пораки, фотографии, геолокациски координати, историја на интернет-пребарување и друго. Со форензичките алатки за мобилните уреди можеме да ги извлечеме податоците на два начини [4]:

- Физичко копирање бит по бит на целата физичка меморија;

- Логичко копирање бит по бит на логичките објекти (датотеки и директориуми).

Разликата е во тоа што кај форензичката копија направена со физичко копирање секогаш можеме да ги обновиме и да ги извлечеме избришаните податоци. Не често сме во можност да направиме физичко копирање поради брзиот развој на технологиите кои се користат во мобилните уреди и претставуваат надмоќ врз форензичките алатки. Понекогаш не сме во можност да ги следиме процедурите и методите, бидејќи имаат свои негативности документирани од (NIST) [12]:

- Исклучувањето на телефонот може да ги активира авторизациските кодови кои се потребни доколку сакаме да пристапиме кон извлекување на податоците;
- Доколку телефонот го оставиме вклучен можно е да се испразни батеријата или поради негова достапност на мрежа можно е да се промени структурата на податоците [4].

1.2. ФОРЕНЗИКА НА GPS УРЕДИ

Форензиката на GPS уредите е релативно нова дисциплина, но претставува наука за извлекување на податоците од GPS уредите. GPS (Global Positioning System) е комплексна технологија, но доколку ја разбереме, може да биде лесна.

GPS е систем за глобално позиционирање. Овој систем содржи 32 сателити кои кружат околу земјината топка и емитираат сигнали кои се достапни до неограничен број на приемници. Вселенскиот (SPACE) сегмент содржи 24 оперативни сателити, поделени по 4 сателити на 6 точно дефинирани распоредени елиптични орбити со наклонетост од 55 степени кон екваторот и на надморска височина од 20.200 километри. Секој сателит кружи околу земјата на секои 11 часа и 58 минути. Четири сателити во орбитата се неправилно

распоредени со цел да се обезбеди континуирана покриеност во случај на откажување на некој од сателитите [5].

GPS сателитите се напојувани преку соларна енергија, но имаат резервни батерии и мали ракетни мотори со цел да се движи секој сателит по својата патека. Сателитите емитираат два радиосигнали L1=1575,42 MHz и L2=1227,6 MHz. Цивилните GPS уреди ја користат L1 фрекфенцијата и овој сигнал содржи информации како:

- Идентификациски код кој го идентификува сателитот кој ја емитира информацијата;
- Ефемерид податоци – содржат информации за состојбата на сателитот, моменталното време и датумот. Овој сигнал е важен за одредување на позицијата;
- Алманах податоци – кои му кажуваат на GPS приемникот каде секој сателит ќе биде во кое било време од денот со цел полесно и побрзо одредување на моменталната геолокациска положба [6].

GPS навигацијата се користи во возила, воздухопловни објекти, бродови, во мобилни уреди и др. Сите овие уреди се со флеш меморија чиј капацитет е голем со што може да се чуваат голем број на податоци од овие уреди. Речиси на сите GPS уреди можат да се извлечат следниве податоци:

- Waypoint – е точка во физичкиот простор и како геолокациска точка е зачувана во GPS уредите. Тоа не значи дека корисникот бил на оваа локација, но со пребарување во уредот за точки од интерес или со внесување точна адреса или геолокациски кординати, локацијата е снимена;
- Route – минимум две геолокациски точки помеѓу кои корисникот се движел со помош на навигација од GPS уредот;
- Log – историја на последни пребарувани точки од интерес;

- Track point – локација зачувана во GPS уредот каде корисникот бил претходно.

Денешните мобилни телефони се со вграден A-GPS чип кој за подобрување на перформансите за побрзо пронаоѓање на моменталната геопозиција, ги користи мрежните ресурси за лоцирање на сателитите. Кај мобилните телефони, GPS сервисите не се користат само за навигација. Можеме да видиме фотографии со геолокациски координати од местото каде е земена фотографијата или уредите на iOS ги зачувуваат информациите од базните станици кои содржат геолокациски податоци како: MCC, MNC, Cell ID, LAC, Latitude и Longitude.

1.3. МРЕЖНА ФОРЕНЗИКА

Со развојот на технологиите, мобилните уреди сè повеќе се наметнуваат како компјутерска платформа во однос на компјутерските работни станици. Брзиот развој на интернет-технологијата, растот на мрежните поврзувања и со брзиот развој на комуникациите онлајн, односно социјалните мрежи каде корисниците можат да разговараат, да споделуваат фотографии и датотеки, проследено е со зголемен број на криминални активности извршени преку интернет. Мрежната форензика е гранка на дигиталната форензика која врши следење, пресретнување, складирање и анализирање на активностите на мрежниот сообраќај со цел откривање на изворот на мрежните напади. Со пресретнувањето и прибирањето на мрежниот сообраќај, ќе откриеме како одреден напад бил извршен преку мрежа или како се случил одреден настан. Мрежните уреди како што се рутерите и серверите се добар извор на мрежни информации, бидејќи рутерите ги насочуваат податочните пакети низ мрежата, а серверите одговараат на сервисните барања. Овие уреди се дел од стандардната мрежна опрема и на нив се наоѓаат сите логови за дата пакетите кои поминале преку нив. Овие рекорди се многу корисни за време на истрагата, бидејќи содржат информации за активностите на корисниците со датуми и со времиња. Со помош на мрежната

форензика идентификуваме од каде е изведен нападот, како е изведен и од кои компјутери бил изведен нападот. Целта е одредување на патеката и изворот на нападот.

2. iPHONE iOS – МОБИЛЕН ОПЕРАТИВЕН СИСТЕМ

Со цел да се пристапи кон форензичка анализа на одреден мобилен уред, важно е да ја разбереме внатрешната работа на самиот уред. Постојат различни уреди на Apple со можност за складирање лични податоци. Конфигурацијата на приспособувањата за iPhone и iPad може да влијае на начинот на кој податоците ќе бидат извлечени. За време на најавата за објавувањето на iPhone, Apple тврдеше дека iOS работи на истото UNIX јадро како MAC OS X. iOS е мобилен оперативен систем на Apple кој потекнува од MAC OS X, со кој ја делат Darwin основата, поради што личи на оперативниот систем UNIX. Оперативниот систем iOS првично бил развиен само за iPhone, но сега се користи и во iPad, iPod и во Apple TV. iOS е дизајниран да биде посредник помеѓу хардверот од уредот и вградените апликации. Има многу сличности помеѓу iOS и Mac OS X, бидејќи јадрото на iOS е базирано на Mac OS X јадрото и содржи неколку слоеви за стартување на апликации. iPhone уредот има хардверски компоненти кои се користат од страна на корисникот за интеракција со уредот, а тоа се: multi touch display и акцелерометар[2]. Корисничкиот интерфејс е базиран на концепт со користење повеќе допирни гестови врз екранот. Контролни елементи на интерфејсот се лизгачот, прекинувачот и неколку копчиња. Со интеракција преку контролните елементи се обезбедува добар интерфејс, бидејќи секој елемент има специфични дефиниции во рамките на iOS. Во iPhone, оперативниот систем е firmware, а не софтвер. Firmware се однесува на сет од програмски инструкции кои овозможуваат хардверот да функционира правилно и софтверот да може да работи со хардверот. Apple објавува нови надградби на firmware со цел подобрување на функционалностите на уредот и поправање на претходните софтверски грешки. iPhone iOS е складиран во внатрешната меморија на уредот и раководи со апликациите во уредот и со повеќето основни функционалности.

[2] Уред кој ја мери силата на забрзување, предизвикано од гравитацијата или поради движење.

Технологиите коишто се користат во iPhone iOS се имплементирани како сет од 4 слоја.

Слика 1. Слоеви на iOS јадрото

Figure 1. iOS core layers

Јадрото на iOS прикажано како на слика 1, според [10] ги содржи следниве слоеви:

- Core OS – овој слој го зафаќа најдолното ниво на iOS и како таков седи директно на врвот на хардверскиот уред. Овој слој обезбедува широк спектар на услуги, вклучувајќи ги low-level networking, пристап до надворешните додатоци и вообичаените основни оперативни системски сервиси како што е управувањето со меморијата и со податочниот систем. Повеќето апликации немаат потреба од услугите од ова ниво, но тие

секогаш се достапни во кој било момент. Како што е претходно наведено, основата на iOS е UNIX јадрото, па поради тоа 6 системски компоненти од овој слој обезбедуваат скоро исти функционалности како UNIX;

- Core Services – овој слој обезбедува основни системски сервиси кои сите апликации ги користат. Без сомнение iOS вклучува и безбедносни можности. Една од компонентите во овој слој е безбедносниот сервис кој се користи за имплементирање за чување на податоците и криптографски функции во keychain[3] базата на податоци на уредот. Содржи основни интерфејси, најчесто C-базирани со цел да овозможи податочен пристап;

- Media – улогата на овој слој е да обезбеди iOS со аудио, видео, анимација и со графички способности. Овој слој се состои од голем број на frameworks[4], кои може да се користат при развивање апликации за iPhone;

- Cocoa Touch – овој слој е на врвот на iOS и содржи рамки кои најчесто се користат од страна на програмерите на апликации за iPhone, за имплементација на визуелниот интерфејс на апликациите. Овој слој е напишан во Objective-C. Оперативниот систем на iPhone користи графички интерфејс кој овозможува корисникот да комуницира со програмите на начин различен од внесување преку тастатурата. Има multi touch interface кој му овозможува на корисникот да работи со уредот користејќи повеќе гестови со прстите врз екранот. Овие два интерфејса го формираат нивото cocoa touch.

2.1. HFSX податочен систем

[3] Keychain е password management system во Mac OS.

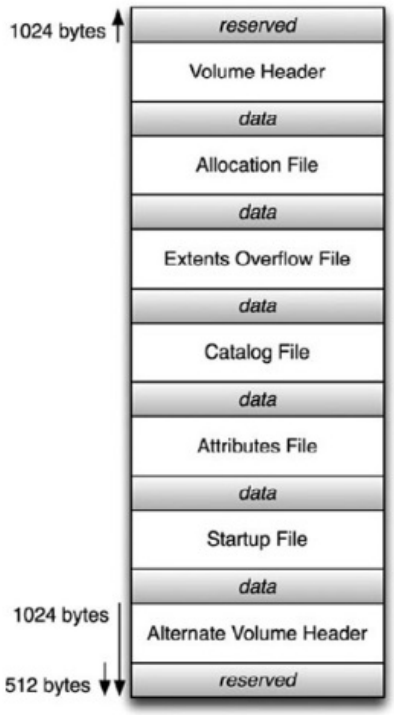
[4] Frameworks се библиотеки кои обезбедуваат интерфејс за развој на апликации за платформата.

Пред да се пристапи кон форензичка екстракција и анализа на iPhone, многу важно е да се разбере како оперативниот систем ги складира во податочниот систем. Податочниот систем што се користи од страна на iPhone според [9] е HFSX, развиен од страна на Apple со цел да се надминат некои од ограничувањата наметнати од страна на HFS (Hierarchical File System) податочниот систем. Apple го развиваше овој податочен систем под кодното името Sequoia. HFSX податочниот систем е идентичен со HFS Plus податочниот систем, но има додатна можност за Case-Sensitive имиња на датотеки, односно повеќе датотеки може да имаат исто име, но со различни големи и мали букви содржани во името на датотеката. Со зголемувањето на големината на меморискиот простор, ограничувањата наметнати од страна на претходниот податочен систем, како што е HFS, тежнееја да станат проблем. HFS Plus волумен форматот е подобрена верзија од HFS, и е дизајниран со цел да се подобрат некои битни карактеристики како што се:

- Ефикасното користење на диск просторот. Во HFS Plus основните алокациски единици се наречени алокациски блокови. За разлика од HFS, кој користи 16-битна вредност за идентификување на алокацискиот блок, додека HFS Plus користи 32-битна вредност за алокацискиот блок. Со ова HFS Plus не дозволува блоковите да бидат споделени помеѓу повеќе датотеки, односно слободниот диск простор прецизно да се дистрибуира меѓу голема количина на датотеки со мала големина. Со повеќето алокациски блокови се добива помалку неискористен простор и помала големина на алокацискиот блок. Алокациските блок броеви се чуваат како 32-битни цели броеви со 4kb како заедничка блок големина. Големината на алокацискиот блок е избрана автоматски со имплементација, во зависност од геометријата на дискот. Блок со големина 8.192 бајти, датотека со должина 24.580 бајти ќе одвои 4 блокови, иако само 4 бајти од последниот блок ќе бидат во употреба [9];

- Поддржува имиња на датотеки во Unicode форма. За разлика од HFS кој користи 31 бит за зачувување на имињата на датотеките, а притоа не зачувувајќи никаква информација за имињата на датотеките поради нивна интерпретација, HFS Plus поддржува имиња на датотеки во уникод формат до 255 знаци, со што се овозможува да се има описно име;
- Метаподатоци - HFS Plus користи датотека за својства во која се зачувуваат информации за датотеки и директориуми и истата датотека ќе биде избришана со бришење на соодветната датотека или директориум. Кај HFS волумен, форматот при бришење на датотека или директориум, датотеката со својства не можеше да се избрише;
- Компатибилност со други оперативни системи, различни од MAC OS X.

Овој податочен систем е составен од голем број објекти кои во комбинација обезбедуваат информации кои се неопходни за управување со податоците на волуменот.



Слика 2. HFS Plus волумен формат

Figure 2. HFS Plus volume format

Секој HFS Plus систем е направен од структури, кои претставуваат дел од физичкиот диск. Секој систем може да биде составен од повеќе волумени, при што секој волумен ќе се состои од: header, alternate header и пет специјални датотеки. Три од овие пет специјални датотеки ја содржат структурата B-tree: Catalog file, Extents overflow file и Atributes file. B-tree е податочна структура која дозволува податоците да бидат ефикасно пребарувани, прегледани, модификувани или отстранети. Се состои од јазли кои ја зачувуваат главната информација со цел побрзо пребарување.

На слика 2 е претставена HFS волумен структурата, содржана од 9 главни структури [13]:

- Првите 1024 бајти се резервирани за блоковите за бутирање на уредот и се наоѓаат на сектор 0 и 1;
- Volume Header/Alternate Volume Header – овие 1024 бајти се наоѓаат на сектор 2 и содржат инфо во однос на структурата на HFS волуменот, како големина на алокациски блок и кои блокови се слободни за користење. Постои и резервна копија од Volume Header која се наоѓа на последните 1024 бајти на HFS волуменот;
- Allocation file – алокациската датотека следи кои алокациски блокови се во употреба од страна на податочниот систем. Форматот на оваа датотека се состои од 1 бит за секој алокациски блок. Ако битот е поставен, блокот е во употреба и обратно, ако битот не е поставен, блокот е достапен за користење;
- Extents overflow file – содржи информации за фрагментираните датотеки, како и за оштетените блокови;
- Catalog file – оваа датотека содржи информации за датотеките и директориумите со цел нивно полесно лоцирање во рамките на волуменот;

- Attributes file - содржи metadata[5] за секој од петте специјални датотеки;
- Startup file – содржи информации за бутирање на уредот;
- Последниот сектор е резервиран за користење од страна на Apple.

Стандардното приспособување за HFS Plus е да користи journaling[6]. Ова претставува механизам со кој сите трансакции на дискот се логирани (запишани) во log датотека. Во случај на неочекувано исклучување, log датотеката може да помогне да се одреди кои промени на дискот биле направени и со тоа функцијата journaling помага во спречување на оштетување на податочниот систем. Секогаш кога оперативниот систем мора да направи некоја промена врз податочниот систем, првин мора таа промена да биде запишана во journal датотеката, па потоа ќе се направи промената врз податочниот систем.

[5] Податоци кои ги опишуваат другите податоци.

[6] Journaling е податочен систем кој ги следи промените во log датотека, пред да ги изврши во главниот податочен систем.

3. BACKUP НА ПОДАТОЦИТЕ ОД iPHONE

Можеме да го користиме Sync програмот iTunes за да направиме логичка копија од податоците на iPhone [15]. Со овој метод нема да ги извлечеме избришаните податоци. За да направиме логичка копија преку iTunes, треба да го имаме главниот компјутер со кој iPhone претходно бил синхронизиран, во спротивно, нема да можеме да ги извлечеме сите податоци од уредот. Ако го синхронизираме уредот со друг компјутер, ќе изгубиме од податоците поради Digital Rights Management. Доколку го имаме главниот компјутер, треба да ги лоцираме датотеките со листа на својства (plist), кои се и сертификати за поврзување (pairing certificates) и се наоѓаат на различни места во зависност од видот на оперативниот систем на работната станица [16]:

- Mac OS X - /private/var/db/lockdown
- Windows XP - C:\Documents an Settings\[username]\Application Data\Apple Computer\Lockdown
- Win Vista - C:\Users\[username]\AppData\roaming\Apple Computer\Lockdown
- Windows 7 - C:\ProgramData\Apple\Lockdown

Откако оваа датотека ќе биде лоцирана и ископирана на форензичката работна станица на истата локација во зависност од оперативниот систем, и со вклучувањето на уредот преку USB, тој ќе се појави како прикачен уред на станицата. По оваа процедура постојат мноштво програми, како за правење image на меморијата така и за вадење на важните кориснички податоци.

Со самото конектирање на iPhone со iTunes, корисникот може да иницира синхронизација помеѓу iTunes и iPhone. Овој процес ќе ги вчита во уредот сите слики, видеа, контакти и апликации кои се чуваат во iTunes. Корисникот има опција да креира backup од телефонот. Автоматски backup се иницира за време на процесот на синхронизација. Backup датотеките се зачувани на одредена локација во зависност од типот на оперативниот систем на работната станица [16]:

- Windows XP - C:\Documents and Settings\<user name>\Application Data

\Apple Computer\MobileSync\Backup\

- Windows Vista/Windows 7 - C:\Users\<user name>\AppData\Roaming\Apple Computer\MobileSync\Backup\
- Macintsoh - Users/<user name>/Library/Application Support/MobileSync/ Backup/

Секој директориум со backup содржи неколку датотеки:

- Status.plist – обезбедува статус за последниот backup;
- Info.plist – содржи инфо за уредот, име на уредот, IMEI, телефонски број и др.;
- Manifest.plist – содржи листа на сите backed up датотеки, нивното време на модификација и hash вредноста.

За сите овие backup датотеки постои SHA 1 hash вредност [16]. iTunes има можност да иницира шифриран backup со внесување лозинка. Доколку backup не е заштитен со лозинка, keychain датотека која содржи username и password, е шифрирана со хардверските клучеви зачувани на iPhone. Ако оваа датотека се обидеме да ја отвориме, ќе можеме да видиме само одредени податоци, но лозинките се шифрирани. Доколку backup е заштитен со лозинка со користење на „Elcomsoft phone password breaker“, ќе ни овозможи форензички пристап до заштитените backups за iPhone. Оваа програма може да го чита и да го дешифрира системот за раководење на лозинките. За користење на овој софтвер, уредот не е потребен, потребни се backup датотеките, особено „Manifest.plist“. Овие синхронизирани податоци може да се извлечат од backup во разбирлива и во читлива форма со помош на софтверот „Iphone backup analyzer“.

4. iPHONE DATA STORAGE

Во мобилната форензичка анализа треба да знаеме кои податоци можат да бидат најдени, каде се складирани податоците и како да дојдеме до нив. iPhone според [1] има внатрешна solid state NAND флеш меморија која е поделена на две партиции. Првата е системската или root партиција (disk0s1s1), чија големина зависи од хардверскиот модел и од верзијата на firmware, и е со големина од 300 MB до 1.5 GB. Во оваа партиција се сместени оперативниот систем (firmware) и вградените апликации. Оваа партиција е секогаш прикачена како read only, со цел да не може да се направат никакви промени врз системот од страна на корисникот или трети апликации. Оваа партиција се прикачува во read-write мод само кога се прави надградба на оперативниот систем. Во случај на jailbreak, оваа партиција цело време е во read-write мод [14], бидејќи се врши промена на AFC[7] сервисот. Втората партиција е корисничката партиција (disk0s1s2), чија големина е целиот останат простор и ги содржи сите кориснички податоци. Корисничката партиција е прикачена во read-write мод како /private/var/. Целта на ваквото системско партиционирање е со цел за време на надградбите врз оперативниот систем, корисничката партиција да остане непроменета.

```
Sasho:~ root# cat /etc/fstab
/dev/disk0s1s1 / hfs rw 0 1
/dev/disk0s1s2 / private/var hfs,nosuid,nodev rw 0 2
```

Слика 3. Преглед на достапните партиции на iPhone

Figure 3. summary of available iPhone partitions

Корисничката партиција ги содржи следниве директориуми:

[7] Apple File Connection сервис кој се извршува во секој iDevice и го користи iTunes за пренос на датотеки.

- **/Var** – ги содржи сите кориснички лозинки во шифрирана форма;
- **/Private** – ги содржи сите податоци до кои корисникот пристапил преку останатите апликации;
- **/Library** – ги содржи телефонскиот адресар, сите логови за телефонски повици, текстуални пораки, податоци од вградениот интернет-пребарувач Safari, Google Maps и др.;
- **/Media** – во овој директориум се содржат сите фотографии и видеоклипови;
- **iTunes_Control** – тука се наоѓаат сите аудио и видеоклипови кои биле синхронизирани со iTunes.

Информациите кои можат се видат на iPhone се трајно запишани на уредот се додека не бидат презапишани. Повеќето од овие податоци се складирани како дел од апликации инсталирани на уредот. Има различни типови на апликации, вклучувајќи ги и тие кои доаѓаат со самиот уред, апликации инсталирани од страна на производителот, апликации инсталирани од страна на корисникот преку iTunes App Store или инсталирани на jailbroken[8] уред преку други извори. Следниве податоци може да бидат извлечени од iPhone:

- Вградени апликациски податоци:
    - Calendar – тука се зачувани сите настани, планирани состаноци со точно дефиниран датум и време;
    - Call Logs – содржи историја на сите влезни, излезни и пропуштени повици. Во зависност од верзијата на софтверот зависи и бројот на рекорди кои ќе бидат зачувани во оваа датотека;
    - Voicemail – се содржат сите преслушани говорни пораки;
    - Text Messages – сите пратени и примени текстуални пораки со точен датум и време, како и телефонскиот број на примачот или испраќачот на пораката;

[8] Процес на отстранување на ограничувањата, со што iOS корисниците добиваат root пристап кон OS.

  - Photos, Videos – сите фотографии и видеоклипови кои се креирани преку вградената камера на iPhone или се импортирани преку електронска пошта или преку iTunes;
  - YouTube – историја на прегледани видеоклипови од YouTube сервисот;
  - Google Maps – сите претходно пребарани геолокациски точки од интерес, како и патеката на движење помеѓу две точки;
  - Voice Memos – говорни белешки;
  - Notes – сите кориснички белешки со датум и време на креирање;
  - Contacts – содржани се сите телефонски контакти со име, презиме, телефонски број, слика, електронска пошта, место на живеење и др.;
  - Web history и bookmarks – сите посетени и омилени веб-страници преку вградениот интернет-пребарувач Safari.
- Податоци за географска локација – GPS координати, слики и видеа со GPS координати;
- Синхронизирани податоци – видеа, песни, слики синхронизирани преку iTunes;
- И-мејл кориснички сметки, како и конфигурациски приспособувања;
- Приспособувања на уредот, wi-fi и Bluetooth инфо, верзија на firmware и софтвер;
- Кориснички имиња, и-мејл адреси, лозинки.

Секогаш кога ќе се инсталира апликација на iPhone, се креира директориум во private/var/mobile/Applications. Името на директориумот е единствен идентификациски стринг кој содржи 32 алфанумерички знаци (0B96B9F4-03BA-4774-B912-508151B10C3E). Овој стринг е различен за секоја нова апликација. Внатре во секој директориум на новите инсталирани апликации постојат 4 различни директориуми и неколку поддиректориуми:

- Documents – се користи за зачувување кориснички документи и податоци за апликацијата во sqlite database формат;
- Library – содржи неколку поддиректориуми и овде не се зачувуваат кориснички податоци. Се користи за зачувување специфични апликациски датотеки:
    - Library/Preferences – содржи датотеки со својства за самата апликација;
    - Library/Caches – содржи податоци симнати преку интернет од самата апликација. Податоците во овој директориум ги брише самата апликација по потреба.
- Tmp – се зачувуваат привремени датотеки;
- Application.app – место каде е апликацијата.

Сите податоци на iPhone се зачувани на внатрешната меморија, бидејќи нема надворешен мемориски слот. iPhone има стандардна директориумска структура каде се зачувани различни датотеки. Партицијата со податоци има многу информации кои ќе помогнат во која било истрага. Следниве директориуми и датотеки за нас како форензичари имаат големо значење:

- /private/var/mobile/Applications – во овој директориум се наоѓаат апликациите инсталирани преку iTunes. Сместени се во поддиректориуми препознатливи по 32 алфанумерички знаци содржани во името на датотеката. Во овој директориум и во сите поддиректориуми ќе најдеме многу важни артефакти содржани во plist[9] датотеки и Sqlite[10] базите на податоци;
- /private/var/Keychains – директориум кој се наоѓа на корисничката партиција во read/write мод. Содржи многу кориснички податоци, но една од поважните датотеки е sql базата на податоци keychain-2.db. Според Apple,

[9] Plist е датотека во која се зачувуваат податоци чија податочна структура или форма првин се преведува во формат кој може да биде зачуван.

[10] Систем за менаџирање со релациски бази на податоци.

keychain е место за безбедно складирање осетливи податоци како: кориснички имиња, лозинки, мрежни лозинки и др. Сите податоци во оваа база на податоци се шифрирани;

- /private/var/mobile/Library – овој директориум содржи датотеки со системски приспособувања, како и податоци за: AddressBook, Notes, Mail, Calendar, Text Messages, Safari содржани во plist и Sqlite датотеки:
    - RecentSearches.plist – датотека во XML формат која содржи зборови кои биле пребарувани преку вградениот интернет-пребарувач Safari за време на користењето на уредот. Локацијата на оваа датотека е /private/var/mobile/Library/Caches/Safari. Оваа база на податоци содржи листа на последни 20 пребарувања;
    - call_history.db – се содржи целата историја на телефонската комуникација. Се наоѓа во /private/var/mobile/Library/CallHistory. Ова е една од трите најголеми бази на податоци и е единствена база на податоци со рестрикции, бидејќи може да чува до 100 рекорди, но има апликации со чија помош се заобиколуваат овие ограничувања и може да се чуваат до 400 рекорди. Во оваа SQLite база на податоци има две табели:
        - SqliteDatabaseProperties;
        - Call.

      Во табелата SqliteDatabaseProperties, има клуч call_history_limit, чија фабричка вредност е 100 и претставува максимален број на ентитети што може да се зачуваат во call_history.db базата на податоци;
    - sms.db – sql база на податоци која ги чува сите примени и испратени текстуални пораки, но, исто така, ги содржи и избришаните пораки. Се наоѓа во /private/var/mobile/Library/SMS. Оваа база на податоци има ограничување на нејзината големина до 15 MB на мемориски простор, што е некаде околу 90.000 текстуални пораки. Доколку се надмине овој лимит, iPhone нема да може да прима нови пораки и на екранот ќе се појави порака како на слика 4:

Слика 4. Порака при надминат лимит на текстуални пораки

Figure 4. Message when the SMS limit exceeded

Доколку на екранот на iPhone се појави оваа порака, ќе мора рачно да избришеме дел од непотребните пораки и да се рестартира уредот. Овие лимитирања се направени со цел да се овозможи ефикасност на базата на податоци;

- AddressBook.sqlitedb – база на податоци со 18 табели во која се содржани сите лични контакти со име, презиме, телефонски број, адреса на живеење, и-мејл, роденден и др. Се наоѓа во /private/var/mobile/Library/AddressBook. Оваа база на податоци нема лимит на максимален број на контакти кои можат да бидат зачувани;
- dynamic-text.dat – во оваа датотека се зачувуваат сите зборови кои биле внесени за време на користењето на уредот, независно преку која било апликација. Се наоѓа во /private/var/mobile/Library/Keyboard. Оваа база на податоци ги зачувува сите зборови коишто корисникот ги внел во текстуалните полиња. Не зачувува зборови внесени во полиња кои се означени како безбедни (secure). Лимитот на оваа база на податоци е околу 600 зборови кои уредот ги користи за автоматско комплетирање на вообичаени предвидливи зборови;
- И-мејл податоци – за разлика од останатите бази на податоци, Protected Index нема екстензија, но всушност е SQLite база на податоци и во неа се зачувани сите добиени, пратени и избришани и-мејл пораки. Се наоѓа во /private/var/mobile/Library/Mail. На оваа локација се содржани сите прикачени датотеки кои биле добиени или

испратени преку и-мејл порака, а претходно биле отворени како датотеки со соодветна апликација. Во директориумот Mail има поддиректориум Messages, во кој се зачувани многу необработени и-мејл пораки со emlx екстензија;

- lockCache_encryptedA.db – со цел вградениот GPS уредот од iPhone да ја добие моменталната точна локација, во оваа база на податоци се зачувуваат GPS координати од секоја базна станица на која телефонот се најавил, како и податоци за најблиските wi-fi уреди. Се наоѓа во /root/Library/Caches/locationd.

- private/var/mobile/media/DCIM – во оваа датотека се наоѓаат сите фотографии и видеоклипови.

Преносните уреди на Apple, своите податоци ги зачувуваат во plist и во sqlite датотеки. Plist датотеките познати и како property list, ги зачувуваат податоците во XML формат. Со користење повеќе типови на објекти ги организира податоците во именувани вредности. Овие датотеки се или во бинарен или во ASCII формат. Датотеката во ASCII формат е лесно читлива со кој било едитор на текст. Често се користат за зачувување кориснички приспособувања. Голем број од корисничките податоци се зачувуваат во sqlite релациски бази на податоци. Релацијата на овие бази на податоци е осмислена за да ни овозможи подобро да се визуелизираат податоците на интерфејсот на iPhone. На пример, во логот на повици, во корисничкиот графички интерфејс (GUI) се покажува името на лицето. Call log базата на податоци врши пребарување во AddressBook базата на податоци за вистинскиот број и името поврзано со овој број и го прикажува името наместо бројот. Кога телефонскиот повик е остварен и бројот не е лоциран во AddressBook базата на податоци, на GUI е прикажан само бројот. SMS базата на податоци е во врска со податоците од AddressBook со имињата и со зачуваните броеви.

5. ФУНКЦИИ НА iPhone

Со објавувањето на iPhone, Apple направи промена на пазарот за мобилни телефони, со што ги натера другите производители со објавувањето нови телефонски уреди да се натпреваруваат со iPhone. Главните функции на првиот iPhone беа телефонска комуникација, пристап на интернет, и-мејл, асистенти со лични податоци, како и календар, адресар и забелешки, поврзување со iTunes и со YouTube. Втората генерација на iPhone телефони имаа повеќе функции како GPS услуги, кои можеа да се поврзат со Google Maps и да му кажат на корисникот која е точната географска локација. Со пристигнувањето на третата генерација на iPhone и со новиот 2.0 софтвер, се овозможи постарите генерации на iPhone да пристапат на App Store. App Store е пазар за бесплатни апликации и апликации со пари.

5.1. РАБОТНИ РЕЖИМИ НА iDevices

iDevices можат да работат во различни оперативни режими на работа. Според Andrew Hoog и Katie Strzempka [1], оперативните режими на работа се следниве:

- Normal mode – кога уредот се вклучува на типичен начин познат како нормален режим. Повеќето од активностите што се извршени на iPhone, се извршени во нормален мод, освен ако поинаку не е одредено;
- Recovery mode – доколку при процес на надградба на оперативниот систем на уредот, процесот не завршил како што треба, уредот можно е автоматски да отиде во овој мод или корисникот треба да го бутира уредот во iBoot, заобиколувајќи го вчитувањето на оперативниот систем. iBoot е фаза на Apple 2 bootloader и е место каде е сместен recovery mode. Овој оперативен режим е потребен за извршување одредени функции како што се активирање на уредот, надградба на iPhone или физичка форензичка аквизиција. За да се влезе во recovery mode првин го гасиме уредот, потоа

се држи притиснато Home копчето и се приклучува преку USB кабелот на компјутер. Home копчето го држиме притиснато сè додека на екранот не се појави „Connect to iTunes“ и потоа се пушта. За време на овој оперативен мод сите податоци на уредот ќе се избришат;

- DFU[11] mode – во овој оперативен мод уредот комуницира со iTunes и не се подига оперативниот систем или boot loader. Овој оперативен режим е потребен за иницирање различни активности, најчесто познати како „Device firmware upgrade“ и „Jailbreak“. За да се влезе во овој режим на работа, iPhone претходно треба да е вклучен на компјутер преку USB кабелот. Копчињата Home и Power се држат заедно 10 секунди. Кога уредот е во DFU режим, екранот ќе биде црн. Излезот од Recovery и DFU режимите на работа се прави со ресетирање на уредот со држење на копчињата Home и Power неколку секунди или со помош на RecBoot апликацијата која може автоматски или да го внесе или да го изнесе уредот во овие модови на работа. Главната разлика помеѓу овој мод и Recovery модот е тоа што првиот се користи за да се инсталира постара верзија на iOS и се заобиколува iBoot, додека кај Recovery модот е обратно.

5.2. БЕЗБЕДНОСТ НА МОБИЛНИТЕ УРЕДИ

Модерните технологии стануваат сè повеќе интегрирани во секојдневниот живот со брзиот раст на популарноста на паметните преносни уреди, автоматски дојде и побарувачката за повеќе и подобри функционалности. Зад нас се деновите кога телефоните беа во големина како цигла, со аналогни копчиња и многу мала меморија. Денешните мобилни телефони и таблети се моќни компјутерски уреди. Освен основните телефонски комуникациски можности, испраќање и примање текстуални пораки и зачувување лични контакти, денешните мобилни уреди се со можност за пристап до интернет, испраќање и

[11] Device Firmware Upgrade – мод кој овозможува уредите да бидат доведени во првобитната состојба или ресетирани на фабричките приспособувања.

примање и-мејл пораки. Многу од уредите имаат вграден GPS со што на корисниците им се овозможува користење на геолокациските сервиси. Побарувањата за повеќе функционалности водат до поголема комплексност на оперативниот систем. Со ова, мобилните уреди се повеќе ранливи на грешки и падови на оперативниот систем, како и на безбедносни пропусти. Паметните преносни уреди, со нивната мала големина, но голема компјутерска моќ, во себе содржат многу лични и осетливи информации со што уредите се изложени на безбедносни ризици:

- Кражба на уредот – мобилните уреди имаат додатен ризик на кражба поради нивната големина и преносливост;
- Злонамерен софтвер – кражба на осетливи информации;
- Јавни wi-fi точки – безжичниот мобилен сообраќај може да не е шифриран и тој може да биде пресретнат.

Производителите треба да се свесни на овие безбедносни пропусти и треба да преземат противмерки со цел зголемување на безбедноста на податоците кои се во самите уреди. Според Apple [7], безбедноста на iOS се дели на 4 слоја:

- Системска архитектура – платформата заедно со хардверот нудат заштита на уредите на iOS. Тука спаѓаат Secure Boot Chain, Code Signing и ASLR;
- Шифрирање и заштита на податоците со хардверски безбедносен механизам со AES 256-битно шифрирање во случај на кражба или на напад;
- Мрежна заштита при пренос на податоци со VPN можност, шифриран wi-fi со WPA2 протокол, SSL, TLS и Bluetooth поддршка со заштита;
- Пристап до уредот – се спречува неавторизиран пристап до самиот уред.

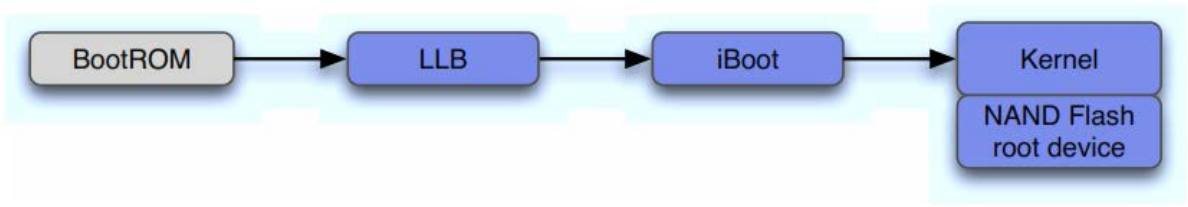
5.3. БЕЗБЕДНОСНИ МЕХАНИЗМИ НА iOS

Мобилните уреди на iOS оперативниот систем се уреди кои постигнаа голем успех на пазарот. Овие уреди станаа дел од секојдневниот живот. Ниту еден

мобилен уред не е целосно сигурен, но iOS има многу добри механизми на заштита, кои функционираат на non-jailbroken уреди на iOS. Уредите на iOS се ранливи на физичко ниво доколку напаѓачот користи пристап или техника како методот на jailbreak. Безбедносните механизми на iOS ги штитат податоците дури и при губење физичка контрола врз уредот.

5.3.1. SECURE BOOT CHAIN

Процесот на подигнување на iPhone има неколку фази на подигнување кои се вчитуваат само доколку моменталната фаза направи успешна потврда на интегритетот и автентичноста на следната фаза со цел таа да биде вчитана и подигната. Редоследот на фазите на подигање е следниот:



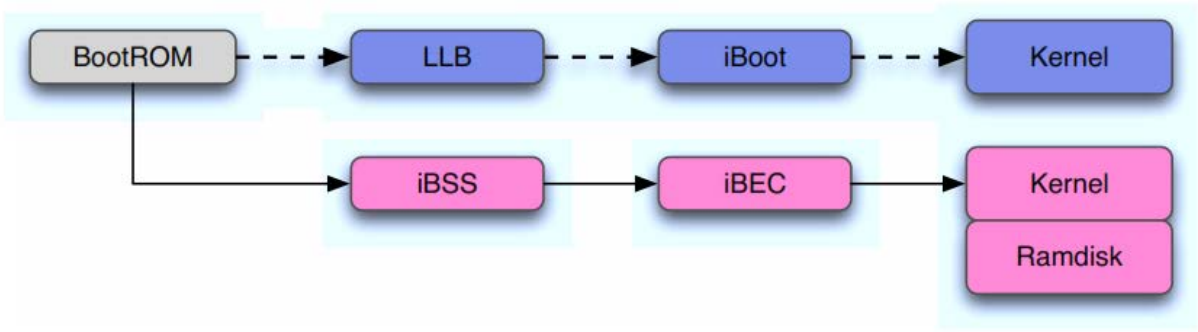
Слика 5. Процес на нормално подигнување на уредот

Figure 5. Normal process of booting up the device

- BootRom (VROM) – виртуелна read-only меморија. BootRom извршува една од двете функции:
  - Подигнување на уредот;
  - DFU мод.
- LLB – Low Level BootLoader, извршува неколку setup рутини и врши проверка на потписот на следното ниво iBoot;
- iBoot – второ ниво на boot loader, го покренува познатиот Recovery мод. Има интерактивен интерфејс кој се користи кога уредот е приклучен преку USB;

- Kernel + NAND Flash – односно оперативниот систем iOS кој е ранлив на напади врз ramdisk.

За време на процесот на подигнување, се гради еден верижен механизам на доверба, поради што е многу тешко да се направи пристап при нормално подигнување на уредот. Со цел да се обезбеди интегритет при целиот boot процес, секоја компонента од boot процесот е криптографски потпишана од Apple, односно кога уредот се подигнува во нормален мод се создава верижен механизам на доверба, кој всушност е серија од проверки на RSA потписите помеѓу софтверските компоненти. Доколку при процесот на подигнување, некоја од софтверските компоненти не може да го верификува следниот чекор, целиот процес на подигнување се стопира и уредот оди во recovery мод. Доколку BootRom не може да го верификува следниот чекор, уредот оди во DFU мод. И во двата случаи, уредот мора да се приклучи на iTunes и да се ресетира на фабричките приспособувања. DFU модот има различна секвенца на подигнување на уредот, споредено со нормалниот мод и тоа:



Слика 6. DFU мод

Figure 6. DFU mode

- BootRom – го подигнува вториот boot loader IBSS;
- IBSS – послаба верзија на iBoot, каде недостигаат работи како што е интеракцијата со податочниот систем. Испраќа порака до iTunes за време на restore процесот, со цел да врши надзор врз целиот процес;

- IBEC – исто како IBSS, послаба верзија на iBoot, врши подготовка и го извршува процесот на restore ramdisk;
- Restore Ramdisk – процес на бришење на NAND меморијата и полнење со OS од firmware датотеката ipsw.

5.3.2. SANDBOX

Со цел да се задржи едноставноста на системот, податочниот систем на iOS е дизајниран со вградени безбедносни механизми. Корисниците на базираните уреди на iOS, немаат пристап до податочниот систем, односно не им е дозволено да креираат и да создаваат нови податоци. На апликациите не им се дозволува целосен пристап до податочниот систем, но оперативниот систем на секоја апликација доделува одвоен податочен систем наречен sandbox[12]. Ова претставува една средина која ги изолира апликациите, не само од другите апликации туку и од оперативниот систем, односно одредена апликација нема пристап до датотеките креирани од друга апликација. Секоја апликација има пристап до содржината на своите датотеки. За време на инсталирањето нови апликации, се креира директориум за секоја апликација со име кое е единствен идентификациски стринг од 32 алфанумерички знаци, во /var/mobile/Applications директориумот. Сè што е потребно за извршување на апликацијата се содржи во новокреираниот директориум. Сите апликации се извршуваат во средина со „mobile“ кориснички привилегии. Со ова се создава една приватна средина со податоци за секоја апликација, каде им е дозволено пристап за читање до media податоците, читање и запис во адресарот, како и неограничен пристап на мрежни конекции. На овој начин се лимитира штетата што би ја направило трето лице како и превенција од напади. Доколку врз уредот се изврши методот на jailbreak, sandbox заштитата е оневозможена.

[12] Сет на добро структурирани контроли за ограничување на пристап на апликацијата до датотеките, параметрите итн.

5.3.3. DATA EXECUTION PREVENTION

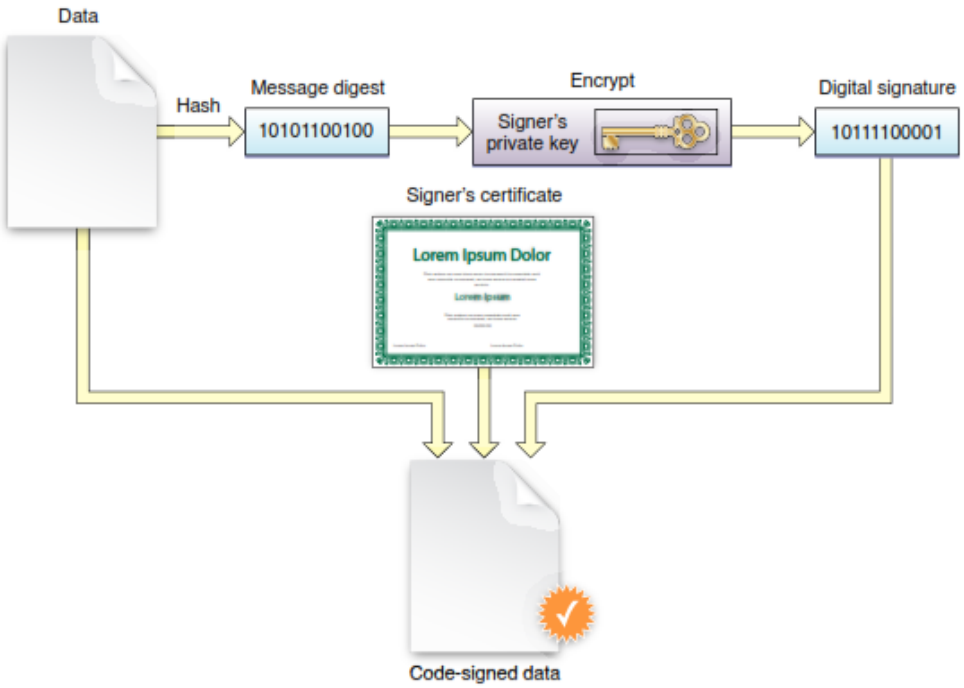
Data Execution Prevention (DEP) е безбедносна особина која е вклучена во речиси сите оперативни системи, чија главна улога е спречување заразени апликации да го извршат кодот кој се наоѓа во меморијата во која не е дозволено извршување. Кога се извршува апликација, се алоцираат мемориски страници за таа апликација, кои содржат маркери, дали овие страници се извршни или не. Доколку одредена страница има извршен маркер на кодот лоциран на одредена мемориска адреса, тој ќе биде извршен и обратно, доколку се стартува код од страна со маркер за неизвршување, процесот од апликацијата се прекинува. Процесорот разликува кој дел од меморијата е со извршен код, а кој дел е со податоци. Со овој безбедносен механизам, меморискиот дел со податоци не смее да содржи код и има привилегии за читање и запишување, а меморискиот простор кој содржи код има привилегии за извршување. Во поновите оперативни системи на iOS се користи безбедносниот механизам на ARM процесорите, наречен XN (Execute Never), со што се ограничуваат привилегиите на мемориските страници. Во исто време, мемориската страница не може да има привилегии за читање, запишување и извршување. Дозволени се само читање и запишување или читање и извршување, со што се обезбедува средина во која било заразена апликација да не може да вметне код во меморијата, кој подоцна би бил извршен. Но, со помош на Return oriented programming[13] (ROP) техниката, може да се заобиколи DEP механизмот. ROP е техника која му овозможува на напаѓачот извршување на код во присуство на безбедносни механизми. Во случај на buffer overflow, кога соодветна функција нема да направи правилна проверка на границите на меморијата до која пристапува, односно функцијата добива повеќе влезни податоци отколку што може да ги зачува, напаѓачот може да внесе одредена количина на податоци во меморијата каде се лоцирани и други променливи. Откако ќе се преземе контролата врз протокот на програмата, соодветна функција (RETN) се извршува без инјектирање на каков било код со цел да се лоцира

[13] Техника која му овозможува на напаѓачот да го изврши кодот во присуство на безбедносните механизми.

адресата на мемориската станица. Кога ќе заврши функцијата (RETN) со извршување, адресата на следниот извршен код се наоѓа на врвот на мемориската станица. Со цел да се развијат безбедносни механизми кога ќе се преполни баферот, податоците кои се наоѓаат во меморијата се маркираат како неизвршни. Но, овој безбедносен механизам во комбинација со code signing механизмот, не дозволува извршување на непотпишан код од страна на Apple, со што може да се спречи извршување на техниката ROP.

5.3.4. CODE SIGNING – потпишување код

Code signing е безбедносен механизам кој спречува неавторизирани апликации да бидат извршени на уредот. Секогаш кога некоја апликација се извршува, се врши валидација на потписот. Се врши дигитално потпишување на извршните скрипти и библиотеки со цел да се потврди дека кодот не бил променет откако дигитално е потпишан.



Слика 7. Дигитално потпишување на извршните скрипти

Figure 7. Digitally signing of the executive scripts

Кернелот нема да дозволи која било апликација да се изврши во iPhone доколку кодот на апликацијата кој се извршува, не е дигитално потпишан од страна на Apple. Кодот е потпишан со користење асиметричен криптосистем и хеш функција. Врз изворниот код се извршува еднонасочна функција со што се хешира изворниот код, а потоа добиениот хеш резултат се шифрира со приватниот клуч на девелоперот и резултира во дигитален потпис. Процесот на потпишување на кодот е составен дел од развојниот процес со што се одржува интегритетот на апликацијата за време на неговата дистрибуција. iOS го проверува дигиталниот потпис на кодот и доколку тој не се совпаѓа, нема да дозволи апликацијата да биде инсталирана и извршена на уредот.

5.3.5. Address Space Library Randomization

Address Space Library Randomization (ASLR) е безбедносен механизам кој за првпат е претставен со појавувањето на iOS 4.3. ASLR е развиен во 2001 година како безбедносна закрпа за системите Linux, а денес се користи скоро во сите оперативни системи. Како додаток на кернелот и во комбинација со претходно споменатите заштитни механизми, треба да обезбеди многу добра заштита од злонамерни кодови. Со користење на ROP, напаѓачот доволно е да знае каде во меморијата е сместена одредена библиотека. ASLR овозможува објектите лоцирани во меморијата да не се на фиксна позиција. Kernel има 256 мемориски локации, секоја со по 2 MB. Со тоа што библиотеките и извршниот код се на непозната мемориска локација, заедно со DEP безбедносниот механизам кој дозволува запис само во делот на меморијата кој не се извршува, го прават iOS отпорен на напади. ASLR се базира на малите шанси напаѓачот да ја погоди точната локација на случајно распоредените мемориски простори.

5.3.6. ШИФРИРАЊЕ И ЗАШТИТА НА ПОДАТОЦИТЕ

Физичката заштита е важен аспект и е предизвик во информатичката безбедност. Бидејќи понекогаш физичката безбедност е многу тешко да се обезбеди, затоа е полесно да се спречи неавторизираниот пристап до податоците. Во базираните уреди на iOS, имплементирани се неколку безбедносни механизми кои ги штитат податоците од неавторизиран пристап и во случај на кражба на уредот. Кај уредите на iOS, заштитата на податоците претставува комбинација на основното хардверско шифрирање заедно со софтверскиот клуч. Уредите содржат AES хардверски процесор кој раководи со шифрирањето заедно со вградените UID и GID клучеви. Сите уреди на iOS се со два вградени клучеви. GID клучот е споделен помеѓу сите уреди од истиот модел и најчесто се користи за дешифрирање на iOS firmware, а UID е единствен за тој уред и претставува хардверски клуч. UID се користи за шифрирање на metadata на податочниот систем, датотеки и keychain датотеки. За UID клучот дури ни самиот производител Apple нема евиденција за кој е клучот и во кој уред е. Овие два клуча не се достапни за нивна екстракција и се користат од страна на криптографскиот алгоритам од AES процесорот за шифрирање и за дешифрирање. Со ова, податоците се криптографски поврзани со соодветниот уред. На овој начин, со отстранувањето на меморијата и поставување друг уред не се врши компромитација на податоците. Хардверската безбедносна карактеристика е користењето на Advanced Encryption Standard (AES) 256-битна криптографска функција и Secure Hash Algorithm (SHA-1) хеш алгоритам, вградени во уредот.

Слика 8. Шифрирање на податоците

Figure 8. Data encryption

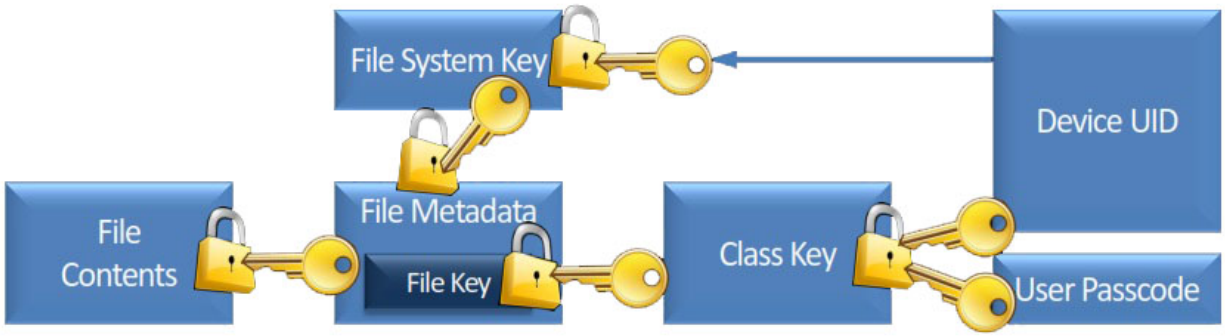
Шифрирањето на податоците е само на корисничката партиција и се врши со хардверски базираниот податочен системски клуч за шифрирање (EMF). Овој клуч се креира автоматски, веднаш по инсталирањето на iOS или по ресетирање на уредот на фабричките приспособувања и е заштитен со UID. Овој клуч се чува на NAND флеш меморија, на локација наречена effacable storage, која се користи за складирање мала количина на податоци со можност за нивно брзо бришење. Концептуалната карактеристика на effacable storage е во случај на брзо бришење на голема количина на податоци со задавање далечинска команда, полесно е да се избрише EMF клучот со кој се шифрирани податоците на уредот. На овој начин, не се врши бришење на корисничките податоци, но тие се некорисни, бидејќи главниот клуч за нивно дешифрирање е избришан. Податоците на уредите на iOS се цело време шифрирани. Доколку некоја од корисничките апликации сака да пристапи до некој податок, тој се дешифрира во лет. Но сепак, трети лица можат да ја пробијат оваа заштита доколку имаат пристап до телефонот. Затоа Apple има воведено дополнителна заштита на податоците на софтверско ниво кое функционира заедно со хардверското шифрирање. Заштитата на податоците или познато под името data protection, е овозможена доколку корисникот го приспособил уредот на iOS да се отклучува со внесување корисничка шифра или passcode. Секогаш кога се заклучува уредот со шифра, уредот ги шифрира датотеките со користење на UID, шифрата и $PBKDF_2$[14] алгоритам и генерира клуч. Овој клуч се чува во меморијата додека уредот е заклучен. При отклучување со користење на шифрата и модификуваната верзија од $PBKDF_2$ алгоритамот, се дешифрира генерираниот клуч. Доколку се совпаѓа уредот ќе се отклучи. За заштита на датотеките се користат 3 клуча и тоа:

- File system key (EMF) – клуч за шифрирање на податочниот систем;
- Class key – зависи од хардверскиот клуч и шифрата;
- File key – се креира за време на создавање на датотеката и зависи од хардверскиот клуч.

[14] (Password-Based Key Derivation Function 2) е клуч за добивање функција која е дел од RSA.

Заштитата на датотеките со клучеви кои се заштитени со таканаречени class клучеви кои произлегуваат од шифрата и AES клучот. За време на развојот на апликациите за уредите на iOS, програмерот за секоја датотека од апликацијата одредува која заштитна класа ќе биде доделена, а ги има 4 и тоа:

- NSFileProtectionNone – без никаква заштита и може да биде пристапено во кое било време:
- NSFileProtectionComplete - со заштита и може да се пристапи само кога уредот е отклучен;
- NSFileProtectionCompleteUnlessOpen – со заштита и може да се пристапи само доколку е отворена датотеката и уредот е отклучен;
- NSFileProtectionCompleteUntilFirstUserAuthentication – со заштита сè додека уредот не се подигне и корисникот ја внесе шифрата за првпат.



Слика 9. Шифрирање на податоците

Figure 9. Data encryption

Целиот процес е прикажан на слика 8 каде при креирањето нова датотека се генерира file key кој е единствен. Овој клуч е зачуван во metadata на датотеката и е заштитен со соодветна класа на клуч. EMF клучот се користи за шифрирање на metadata. Class клучот е заштитен со UID и со корисничката шифра. Можеме да видиме дека шифрата не само што се користи за да се отклучи уредот туку се користи заедно со UID за генерирање клучеви кои се користат при шифрирање и

дешифрирање. Кога уредот е заклучен со шифра, EMF клучот е недостапен и податоците се шифрирани.

6. МЕТОДИ НА ИСТРАЖУВАЧКАТА РАБОТА

Во овој магистерски труд ќе се истражуваат форензичките пристапи врз Apple iPhone 4. Ќе го истражуваме пристапот на физичко извлекување на податоците од корисничката партиција со користење на iOS open source форензичките алатки познати како iPhone Data Protection [18]. Единствен уникатен начин за физичко извлекување на податоците од iPhone уредите е развиен од страна на Jonathan Zdziarski [17]. Овој човек важи за експерт во областа на iOS дигиталната форензика и безбедност. Сите негови форензички методологии кои се користат денес се валидирани од страна на Американскиот меѓународен институт за правда. Во однос на [8] каде сите процеси се автоматизирани и се користат софтверски алатки достапни само за органите за спроведување на законот, тука е презентиран целиот процес од самото создавање форензички услови со приспособувањето на форензичката работна станица и инсталирање алатки кои се достапни за секого.

Истражувањата ќе бидат извршени во прифатливи форензички услови, а притоа внимавајќи да не направиме никаква промена врз податочниот систем со што ќе го запазиме основното правило при процесот на форензичко извлекување. Поради тоа што мобилната форензика нема стандарден процес на пристап кон извлекување податоци како компјутерската форензика, односно не можеме да примениме writeblocker, а постојат многу форензички алатки за кои не знаеме како функционираат, многу е битно податоците кои се наоѓаат на мобилните уреди да не бидат променети. Ова ќе го постигнеме со користење програми кои ги следат сите податоци кои се префрлаат помеѓу мобилниот уред и форензичката работна станица.

6.1. ПРИСПОСОБУВАЊЕ НА ПОТРЕБНАТА ОПРЕМА И ФОРЕНЗИЧКИТЕ OPEN SOURCE АЛАТКИ

Пред да пристапиме кон форензичко извлекување на корисничките податоци од iPhone мобилниот уред, најпрвин треба да подготвиме форензичка средина,

односно работна станица на која ќе биде инсталиран оперативниот систем MAC OS X и ќе бидат инсталирани следниве алатки кои се неопходни за пристап кон логичко извлекување на податоците.

- iOS firmware е оперативниот систем на уредите на iOS кој ги обезбедува сите вградени функционалности. Потребно е претходно да ги дознаеме, како хардверскиот модел на уредот така и софтверската верзија на firmware од уредот кој е предмет на истражување. Откако ќе биде установен соодветниот модел и верзијата на софтверот, можеме да симнеме firmware [19];
- Xcode – е на Apple развојна околина за креирање апликации за Mac, iPhone и iPad. Xcode е сет од развојни софтверски алатки и библиотеки. При инсталирањето на Xcode се инсталира поддршка само за моменталната последна верзија на iOS, а тоа е iOS 7. Бидејќи ние ќе работиме на iPhone уред со iOS 5, ќе мора ги симнеме SDK библиотеките за iOS 5 и да ги инсталираме рачно со следниве наредби:

- sudo cp -R /Volumes/Xcode/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/iPhoneOS5.1.sdk /Developer/Platforms/iPhoneOS.platform/Developer/SDKs/
- sudo cp -R /Volumes/Xcode/Xcode.app/Contents/Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/iPhoneSimulator5.1.sdk /Developer/Platforms/iPhoneSimulator.platform/Developer/SDKs/
- sudo cp -R /Volumes/Xcode/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/DeviceSupport/5.1\ \(9B208\) /Developer/Platforms/iPhoneOS.platform/DeviceSupport/
- sudo rm -f /Developer/Platforms/iPhoneOS.platform/DeviceSupport/Latest

- cd /Developer/Platforms/iPhoneOS.platform/DeviceSupport/
- sudo ln -s ./5.1\ \(9B208\) ./Latest

Откако ќе се инсталира Xcode, потребно е да се инсталираат алатките за командната линија со цел да се овозможи функционирањето на GCC компајлерот.

- Ldid – како што е претходно наведено, еден од безбедносните механизми на iOS е кодот којшто се извршува мора да биде дигитално потпишан. Со цел да се заобиколи овој безбедносен механизам, се користи алатката ldid, која генерира SHA1 хешови, кои се проверуваат од iOS кернелот [20];
- OSXFuse – е развојна алатка со библиотеки за развој на OS X податочните системи. Со оваа алатка можеме да имплементираме функционален податочен систем во корисничкиот простор. Алатката е преземена од [21];
- Mercurial – е алатка која овозможува сет на командни линии и таа е преземена од [22];
- Python – е моќен динамички програмски јазик. Моментално, потребно е да се инсталира верзија 2.7. Се користи за креирање на кернелот;
- PyCrypto – сет на хеш функции како SHA256 и различни шифрирани алгоритми AES, DES, RSA и др.;
- RedSn0w – е алатка развиена од iPhone Dev Team, која врши отстранување на ограничувањата поставени од Apple на уредите на iOS. Овој процес е познат под името jailbreaking. Со оваа алатка се овозможува root пристап до оперативниот систем iOS и можност за инсталирање апликации кои не се симнати од App Store. Со помош на оваа алатка, можеме да ги зачуваме SHSH blobs, односно дигиталните потписи кои се обезбедуваат од Apple за точно познатиот уред и се користат за авторизација на инсталација на соодветна верзија на iOS. На овој начин, можеме да направиме и restore на пониска верзија на iOS, доколку ги имаме соодветните SHSH blobs.

6.1.1. ИДЕНТИФИКАЦИЈА НА ХАРДВЕРСКИОТ МОДЕЛ

Најлесен начин за идентификација на хардверот според Jonathan Zdziarski [8] е да обрнеме внимание на моделот кој е напишан на задната страна на уредот како на слика 10.

Слика 10. Идентификација на хардверскиот модел

Figure 10. Identifying the hardware model

Од слика 10, можеме да видиме дека за моделот A1332 станува збор за iPhone 4 GSM. Според следнава листа, можеме да видиме дека бројот на моделот е идентификација за типот на iPhone, и тоа:

- A1428 – iPhone 5 GSM (standard GSM model in USA for AT&T, T-Mobile, etc.);

- A1429 – iPhone 5 GSM & CDMA (normal CDMA model in USA, Verizon, Sprint, etc.);
- A1442 – iPhone 5 CDMA China;
- A1387 – iPhone 4S, CDMA & GSM;
- A1431 – iPhone 4S GSM China;
- A1349 – iPhone 4 CDMA;
- A1332 – iPhone 4 GSM;
- A1325 – iPhone 3GS China;
- A1303 – iPhone 3GS (GSM only);
- A1324 – iPhone 3G China;
- A1241 – iPhone 3G (GSM only);
- A1203 – iPhone (Original model, GSM only).

6.1.2. ИДЕНТИФИКАЦИЈА НА СОФТВЕРОТ

Пред да пристапиме кон подготовка на форензичката екстракција на податоците, мора да знаеме која е моменталната верзија на firmware на уредот на iOS од кој треба да се извлекуваат податоците. Верзијата на firmware на уредот на iOS можеме да ја дознаеме доколку го вклучиме уредот и ќе можеме да ги прочитаме овие параметри. Но, на овој начин, можеме да направиме уништување на податоците што е спротивно од форензичките правила. Затоа ќе ја користиме алатката BootRa1n, преземена од [23]. Оваа алатка е open source и знае да комуницира со iBoot и IBSS преку low level USB протоколи. Со извршување на командата преку терминалот, bootra1n –s го добиваме следново:

```
=====================================
::
:: iBoot for n90ap, Copyright 2011, Apple Inc.
::
::    BUILD_TAG: iBoot-1219.62.16
::
::    BUILD_STYLE: RELEASE
::
::    USB_SERIAL_NUMBER: CPID:8930 CPRV:20 CPFM:03 SCEP:02 BDID:00
ECD:000002E61E105D0C IBFL:03 SRNM:[88041C16A4S]
```

```
::
====================================
```

Доколку на интернет пребараме за `BUILD_TAG: iBoot-1219.62.16,` ќе видиме дека за таа верзија на iBoot, е iOS 5.1.1 (9B208).

6.2. МЕТОД НА ИЗВЛЕКУВАЊЕ ПОДАТОЦИ

6.2.1. ИНСТАЛИРАЊЕ НА OPEN SOURCE АЛАТКИ

За целта на овој магистерски труд, извршено е истражување врз уред на iOS со следниве спецификации:
DeviceClass: iPhone
ProductType: iPhone3,1 (iPhone 4)
ProductVersion: 5.1.1
HardwareModel: N90AP
FirmwareVersion: iBoot-1219.62.16
BuildVersion: 9B208
SerialNumber: 88041C16A4S
UDID: cfcd0d83f73edfff615ced1aab4f0b3f47847ddb

За да создадеме сопствен ramdisk, потребно е претходно наведените open source алатки од поглавје 5.1. да ги инсталираме на форензичката работна станица со рачни команди преку терминалската командна линија со следниов редослед:

- Вршиме преземање на алатката ldid преку интернет на локалниот диск, доделуваме извршни права на алатката и ја преместуваме во /usr/bin/ директориум;

```
curl -O http://networkpx.googlecode.com/files/ldid
chmod +x ldid
sudo mv ldid /usr/bin/
```

- Ја преземаме најпоследната верзија на развојната алатка OSXFuse преку интернет и таа ја инсталираме:

```
localhost:~ neda$ curl -O -L
https://github.com/downloads/osxfuse/osxfuse/OSXFUSE-2.6.2.dmg
   % Total      % Received % Xferd   Average Speed    Time      Time
Time   Current
                                                 Dload   Upload    Total
Spent      Left   Speed
100    133   100    133      0       0      108         0   0:00:01
0:00:01 --:--:--    184
100 4719k   100 4719k      0       0    203k         0   0:00:23   0:00:23
--:--:--   348k
localhost:~ neda$ hdiutil mount OSXFUSE-2.3.4.dmg
Checksumming Gesamte Disk (Apple_HFS : 0)â€¦
.....................................................................
.............................
            Gesamte Disk (Apple_HFS : 0): verified    CRC32 $D1B1950D
verified    CRC32 $09B79725
/dev/disk2
      /Volumes/FUSE for OS X
localhost:~ neda$ sudo installer -pkg /Volumes/FUSE\ for\ OS\ X/Install\
OSXFUSE\ 2.6.2.pkg -target /
localhost:~ neda$ hdiutil eject /Volumes/FUSE\ for\ OS\ X/
"disk2" unmounted.
"disk2" ejected.
```

- Потребно е да инсталираме неколку python скрипти и со следнава команда се инсталираат автоматски потребните скрипти:

```
localhost:~ neda$ sudo easy_install M2crypto construct progressbar
setuptools

Searching for M2crypto

Best match: M2Crypto 0.21.1

Processing M2Crypto-0.21.1-py2.7-macosx-10.7-intel.egg

M2Crypto 0.21.1 is already the active version in easy-install.pth


Using /Library/Python/2.7/site-packages/M2Crypto-0.21.1-py2.7-macosx-10.7-
intel.egg

Processing dependencies for M2crypto

Finished processing dependencies for M2crypto

Searching for construct

Best match: construct 2.06

Processing construct-2.06-py2.7.egg

construct 2.06 is already the active version in easy-install.pth
```

```
Using /Library/Python/2.7/site-packages/construct-2.06-py2.7.egg

Processing dependencies for construct

Finished processing dependencies for construct

Searching for progressbar

Best match: progressbar 2.3

Processing progressbar-2.3-py2.7.egg

progressbar 2.3 is already the active version in easy-install.pth


Using /Library/Python/2.7/site-packages/progressbar-2.3-py2.7.egg

Processing dependencies for progressbar

Finished processing dependencies for progressbar

Searching for setuptools

Best match: setuptools 0.6c12dev-r88846

setuptools 0.6c12dev-r88846 is already the active version in easy-
install.pth

Installing easy_install script to /usr/local/bin

Installing easy_install-2.7 script to /usr/local/bin


Using
/System/Library/Frameworks/Python.framework/Versions/2.7/Extras/lib/python

Processing dependencies for setuptools

Finished processing dependencies for setuptools

localhost:~ neda$ sudo ARCHFLAGS='-arch i386 -arch x86_64' easy_install
pycrypto

Searching for pycrypto

Best match: pycrypto 2.4.1

Processing pycrypto-2.4.1-py2.7-macosx-10.7-intel.egg

pycrypto 2.4.1 is already the active version in easy-install.pth


Using /Library/Python/2.7/site-packages/pycrypto-2.4.1-py2.7-macosx-10.7-
intel.egg
```

```
Processing dependencies for pycrypto

Finished processing dependencies for pycrypto
```

- Преземање и инсталирање на алатката mercurial која ќе ни овозможи користење сет на командни линии;
- Го преземаме извршниот код и тој е зачуван во директориум iPhone data protection; кој се наоѓа во домашниот директориум на корисникот:

```
localhost:~ neda$ hg clone https://code.google.com/p/iphone-dataprotection/
warning: code.google.com certificate with fingerprint
b7:b5:5e:c9:1b:43:93:f7:17:7c:d7:c2:9d:ad:31:cb:0c:11:c2:de not verified
(check hostfingerprints or web.cacerts config setting)
destination directory: iphone-dataprotection
requesting all changes
adding changesets
adding manifests
adding file changes
added 72 changesets with 2033 changes to 1865 files
updating to branch default
152 files updated, 0 files merged, 0 files removed, 0 files unresolved
localhost:~ neda$ cd iphone-dataprotection
```

- Правиме компајлирање на img3fs.c со цел оваа скрипта да ја употребиме за дешифрирање и шифрирање на ramdisk и кернелот:

```
localhost:iphone-dataprotection neda$ make -C img3fs/
gcc -o img3fs img3fs.c -Wall -lfuse_ino64 -lcrypto -
I/usr/local/include/osxfuse || gcc -o img3fs img3fs.c -Wall -losxfuse_i64 -
lcrypto -I/usr/local/include/osxfuse
14 warnings generated.
```

- Откако претходно сме го идентификувале хардверскиот модел на iPhone и софтверската верзија на firmware, го преземаме соодветниот firmware и тој го зачувуваме во iPhone data protection директориумот;
- Ја преземаме RedSn0w позната алатка по jailbreak процесот на отстранување на ограничувањата на уредите на iOS и нејзината можност за креирање на сопствен iOS firmware:

```
localhost:iphone-dataprotection neda$ curl -O -L
https://sites.google.com/a/iphone-dev.com/files/home/redsn0w_mac_0.9.15b3.zip
   % Total      % Received % Xferd   Average Speed    Time      Time       Time
Current
                                                Dload   Upload    Total    Spent
Left   Speed
100    274      0    274      0       0      499         0 --:--:-- --:--:-- --:--
:--   1602
100    568     0    568      0       0      839         0 --:--:-- --:--:-- --:--
:--       0
100 17.1M   100 17.1M      0       0    308k         0   0:00:56   0:00:56 --:--:-
-   274k
localhost:iphone-dataprotection neda$ unzip redsn0w_mac_0.9.15b3.zip
```

```
Archive:   redsn0w_mac_0.9.15b3.zip
    creating: redsn0w_mac_0.9.15b3/
   inflating: redsn0w_mac_0.9.15b3/boot-ipt4g.command
  inflating: redsn0w_mac_0.9.15b3/credits.txt
   inflating: redsn0w_mac_0.9.15b3/license.txt
   inflating: redsn0w_mac_0.9.15b3/README.txt
    creating: redsn0w_mac_0.9.15b3/redsn0w.app/
    creating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/Info.plist
    creating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/bn.tar.gz
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/bootlogo.png
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/bootlogox2.png
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/corona-A5.tar
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/Cydia.tar.lzma
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/Keys.plist
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/progresslogo.png
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/rd.tar
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/redsn0w
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/rocky-racoon.tar
 extracting: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/PkgInfo
    creating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/Resources/
   inflating: redsn0w_mac_0.9.15b3/redsn0w.app/Contents/Resources/redsn0w.icns
localhost:iphone-dataprotection neda$ cp
redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/Keys.plist .
```

6.2.2. СОЗДАВАЊЕ СОПСТВЕН KERNEL И RAMDISK

Со помош на претходно инсталираните python скрипти, вршиме извлекување на кернелот од веќе претходно преземениот iOS firmware и негово дешифрирање. Откако тој ќе биде дешифриран и распакуван, се вршат неколку закрпи, меѓу кои и проверката на потписите во кернелот. Откако ќе се имплементираат закрпите, се креира нова датотека со веќе имплементираните закрпи.

```
localhost:iphone-dataprotection neda$ python python_scripts/kernel_patcher.py
iPhone3,1_5.1.1_9B208_Restore.ipsw
Decrypting kernelcache.release.n90
Unpacking ...
Doing CSED patch
Doing getxattr system patch
Doing nand-disable-driver patch
Doing task_for_pid_0 patch
Doing IOAES gid patch
Doing AMFI patch
Doing _PE_i_can_has_debugger patch
Doing IOAESAccelerator enable UID patch
Patched kernel written to kernelcache.release.n90.patched
Decrypting 038-5512-003.dmg
Created script make_ramdisk_n90ap.sh, you can use it to (re)build the ramdisk
```

Со помош на развојните алатки SDK и библиотеки за моменталната верзија на iOS 5.1, го создаваме сопствениот ramdisk. Ramdisk е датотека која ја содржи основната верзија на iOS која се користи за преинсталирање на уредот со соодветна iOS верзија, преземена преку iTunes. Ramdisk е основата за пристап до партициите на iPhone. Во оваа датотека се содржат неколку основни датотеки преземени од соодветниот iOS firmware и со промена на /sbin/launchd, ќе се бутираат скрипти кои ќе ги прикачат партициите на iOS во мод за читање и запишување. Ramdisk датотеката во себе содржи SSH сервер. Наједноставна споредба на ramdisk е со Linux Live CD со чија помош правиме подигнување на оперативен систем од цедето и прикачување на корисничките партиции.

```
localhost:iphone-dataprotection neda$ sh ./make_ramdisk_n90ap.sh
Found iOS SDK 5.1
ln -s /System/Library/Frameworks/IOKit.framework/Versions/Current/Headers IOKit
clang -arch armv7 -Wall -isysroot
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/
SDKs/iPhoneOS5.1.sdk/ -DHGVERSION="\"70a150167027\"" -O3 -I. -framework
CoreFoundation -framework IOKit -framework Security -miphoneos-version-min=4.0 -o
device_infos device_infos.c device_info.c IOAESAccelerator.c
AppleEffaceableStorage.c AppleKeyStore.c bsdcrypto/pbkdf2.c bsdcrypto/sha1.c
bsdcrypto/key_wrap.c bsdcrypto/rijndael.c util.c IOKit.c registry.c
ioflash/ioflash.c kernel_patcher.c
device_infos.c:9:28: warning: initializing

codesign -s - --entitlements tfp0.plist device_infos
clang -arch armv7 -Wall -isysroot
/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/
SDKs/iPhoneOS5.1.sdk/ -DHGVERSION="\"70a150167027\"" -O3 -I. -framework
CoreFoundation -framework IOKit -framework Security -miphoneos-version-min=4.0 -o
restored_external restored_external.c device_info.c remote_functions.c
plist_server.c AppleKeyStore.c AppleEffaceableStorage.c IOKit.c
IOAESAccelerator.c util.c registry.c AppleKeyStore_kdf.c bsdcrypto/pbkdf2.c
bsdcrypto/sha1.c bsdcrypto/rijndael.c bsdcrypto/key_wrap.c ioflash/ioflash.c
kernel_patcher.c
restored_external.c:38:33: warning: implicit declaration of function
'IOUSBDeviceDescriptionCopyInterfaces'
         is invalid in C99 [-Wimplicit-function-declaration]
      CFArrayRef usb_interfaces = IOUSBDeviceDescriptionCopyInterfaces(desc);
codesign -s - --entitlements tfp0.plist ioflashstoragekit
Downloading ssh.tar.gz from googlecode
   % Total      % Received % Xferd   Average Speed    Time      Time       Time
Current
                                                Dload   Upload    Total    Spent
Left   Speed
100 3022k   100 3022k      0       0    322k         0   0:00:09   0:00:09 --:--
:--   231k
Archive:   iPhone3,1_5.1.1_9B208_Restore.ipsw
```

```
   inflating: 038-5512-003.dmg
TAG: TYPE OFFSET 14 data_length:4
TAG: DATA OFFSET 34 data_length:107b000
TAG: SEPO OFFSET 107b040 data_length:4
TAG: KBAG OFFSET 107b05c data_length:38
KBAG cryptState=1 aesType=100
TAG: KBAG OFFSET 107b0a8 data_length:38
TAG: SHSH OFFSET 107b10c data_length:80
TAG: CERT OFFSET 107b198 data_length:794
Decrypting DATA section
Decrypted data seems OK : ramdisk
/dev/disk3
      /Volumes/ramdisk
bin/: Already exists
bin/cat: Already exists
bin/launchctl: Already exists
bin/ln: Already exists
bin/mkdir: Already exists
bin/mv: Already exists
bin/rm: Already exists
etc/: Already exists
sbin/: Already exists
usr/: Already exists
usr/bin/: Already exists
usr/lib/: Already exists
var/: Already exists
tar: Error exit delayed from previous errors.
"disk3" unmounted.
"disk3" ejected.
You can boot the ramdisk using the following command (fix paths)
redsn0w -i iPhone3,1_5.1.1_9B208_Restore.ipsw -r myramdisk_n90ap.dmg -k
kernelcache.release.n90.patched
Add -a "-v rd=md0 nand-disable=1" for nand dump/read only access
```

6.2.3. ВЧИТУВАЊЕ НА СОПСТВЕНИОТ RAMDISK ВО iPHONE

За да можеме да ги вчитаме претходно креираните ramdisk и kernel, потребно е мобилниот уред iPhone да го ставиме во DFU мод. За да се влезе во овој режим на работа, уредот iPhone претходно треба да е вклучен на компјутер преку USB кабелот. Најпрвин телефонот се исклучува и откако екранот ќе стане црн, првин се држи притиснато копчето за power сè додека не се појави на екран логото на Apple. Веднаш кога ќе се појави логото на Apple, без да се отпушти power копчето, се притиска копчето home и копчињата home и power се држат заедно 10 секунди. Откако ќе поминат 10 секунди, се пушта копчето power, а се држи копчето home уште 4 секунди. Кога уредот е во DFU режим, екранот ќе биде црн. Откако уредот

ќе биде ставен во DFU режим на работа со помош на алатката RedSn0w, вршиме вчитување на претходно креираниот сопствен ramdisk и kernel со следнава командана линија:

```
./redsn0w_mac_0.9.15b3/redsn0w.app/Contents/MacOS/redsn0w -i
iPhone3,1_5.1.1_9B208_Restore.ipsw -r myramdisk_n90ap.dmg -k
kernelcache.release.n90.patched
```

Откако ќе се изврши оваа командна линија, целиот процес се извршува автоматски од страна на алатката RedSn0w. Откако ќе се заврши со процесот на вчитување на екранот на iPhone, ќе можеме да видиме како се стартува вчитаниот ramdisk и наместо почетното лого се појавува целиот лог на подигнување на ramdisk. Доколку на екранот на iPhone се појави OK пораката, значи дека успешно е вчитан и е покренат ramdisk.

6.2.4. ВОСПОСТАВУВАЊЕ КОМУНИКАЦИЈА

Откако сме завршиле со процесот на вчитување и подигнување на сопствениот ramdisk, потребно е да ја воспоставиме комуникацијата помеѓу iPhone и MAC OS X работната станица. Обично iTunes комуницира со уредите на iOS преку usbmux протокол кој овозможува повеќе конекции преку еден кабел, односно создава TCP конекции со одредена бројка на порти. За целта на ова истражување, потребно е да се овозможи користење две конекции со различни порти. Најпрвин, потребно е да се воспостави комуникација преку криптографски мрежен протокол за безбедносен пренос на податоци познат како SSH, потребно е да се отвори портата 22 и таа да се изрутира кон порта 2222. Знаејќи дека сите податоци на уредот се шифрирани, потребно е да се отвори втора порта 1999, која ќе се користи само за командната линија преку која ќе се изврши дешифрирање на шифрираните податоци во iPhone. Воспоставувањето на комуникацијата и отворањето на портите се вршат со следнава команда:

```
localhost:iphone-dataprotection neda$ python usbmuxd-python-client/tcprelay.py -t
22:2222 1999:1999

Forwarding local port 2222 to remote port 22

Forwarding local port 1999 to remote port 1999
```

Мобилниот уред iPhone е веднаш достапен преку SSH протоколот со **root** корисничко име и лозинка **alpine.** Можеме веднаш да му пристапиме кон воспоставување на комуникацијата:

```
localhost:iphone-dataprotection neda$ ssh -p 2222 root@localhost
```

6.2.5. ПРОБИВАЊЕ НА КОРИСНИЧКАТА ЛОЗИНКА НА iPHONE

Доколку на уредот на iOS е конфигурирано користење на опцијата passcode, потребно е да се направи пробивање на кодот на iPhone за пристап со помош на скрипта која врши brute force напад:

```
localhost:iphone-dataprotection neda$ python python_scripts/demo_bruteforce.py

Connecting to device : cfcd0d83f73edfff615ced1aab4f0b3f47847ddb

Keybag UUID : af0f8631fb88466784f0b42c86b68d93

Enter passcode or leave blank for bruteforce:

Trying all 4-digits passcodes...

0 of 10000 ETA:   --:--:--

10 of 10000 ETA:   0:30:20                                          |

20 of 10000 ETA:   0:30:06                                          |

1800 of 10000 ETA:   0:24:26                                      |

1810 of 10000 ETA:   0:24:24                                      |

10000 of 10000 Time: 0:05:24                                    |

100% |##############################################|

BruteforceSystemKeyBag : 0:05:24.139018

{'passcode': '1810', 'passcodeKey':
'ff0600fe2d6f88ded6c4e9878afb98faef98ffd1919bf0d7a6cfe4e837c0d7a9'}
```

```
True

Keybag type : System keybag (0)

Keybag version : 3

Keybag UUID : af0f8631fb88466784f0b42c86b68d93

-------------------------------------------------------------------------------
-----------------------------------------------

Class                                                                    WRAP
Type          Key
Public key

-------------------------------------------------------------------------------
-----------------------------------------------

NSFileProtectionComplete                                          3      AES
cf7004bc0725f473e0728e9ea183c1dae176df0699a2618fc57e7a646e5a99f1

NSFileProtectionCompleteUnlessOpen                                3      Curve25519
af20194060132e202bea26e437119e758fc7c5d7e2f2712bf5e0407dda301ce3
9913772d287d315775e2c7d732c8f5147c46f9c7298123f6619225a29dd6f465

NSFileProtectionCompleteUntilFirstUserAuthentication 3      AES
de695f354889d93dd5933f31d6e8d855293efa52286888a500f1f67bf73a7d34

NSFileProtectionRecovery?                                         3      AES
afa37fb78b6f4b17f08d0f5db4790ae41c198e2e5efaebbb0c50324a9da7b0a7

kSecAttrAccessibleWhenUnlocked                                    3      AES
8e3ca278221e172582f9c1a9f3081e4a920e928020da615d4eeca287abb8830e

kSecAttrAccessibleAfterFirstUnlock                                3      AES
80bfbf7b9b0d76372ca8f2e7603ad24582c3800e33fca9d60eea16854ab32b22

kSecAttrAccessibleAlways                                          1      AES
b5b8cdaeefd69fdf273a84543f2098e3dbc0d62488c754c345942e1ad8d413aa

kSecAttrAccessibleWhenUnlockedThisDeviceOnly              3      AES
291f3b8c66c399393404937cb5ff1d02fb14b996b85324820f766ca0c0b44169

kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly       3      AES
dd6787b79d9d2270adf33807e833fddd0770b9b66e0cec369a1d2509ce8c52fd

kSecAttrAccessibleAlwaysThisDeviceOnly                            1      AES
26643b551d749d21ffe5162a608fbcde5730a1868e2c2b924c128d3deb0f0e28

Downloaded keychain database, use keychain_tool.py to decrypt secrets
```

Оваа скрипта врши brute force напади, претпоставувајќи дека шифрата содржи 4 бројки и нападот го започнува со 0000, 0001, па сè до 9999. На моментално користениот iPhone уред, од погоре наведениот резултат од извршувањето на скриптата за пробивање на кодот за пристап, можеме да видиме дека пристапниот

код е 1810. Од табела 1, можеме да споредиме колку време би ни требало доколку кодот не содржи во себе само 4 бројки.

Сложеност на кодот	Потребно време за пробивање
4 бројки	18 минути
4 алфанумерички знаци	51 час
5 алфанумерички знаци	8 години
8 алфанумерички знаци	13000 години

Табела 1. Потребно време за пробивање на кодот

Tabel 1. Time needed to reveal the code

6.2.6. МЕТОД НА ФИЗИЧКО ИЗВЛЕКУВАЊЕ НА ПОДАТОЦИТЕ

Откако сме воспоставиле комуникација со уредот, сме го пробиле кодот за пристап, пристапуваме кон користење на методот за физичко извлекување на податоците од iPhone. Најпрвин, доделуваме извршни права на скриптата за копирање на податоците и потоа таа ја извршуваме преку командната линија:

```
localhost:iphone-dataprotection neda$ chmod +x dump_data_partition.sh
localhost:iphone-dataprotection neda$ ./dump_data_partition.sh
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
root@localhost's password:
Device UDID : cfcd0d83f73edfff615ced1aab4f0b3f47847ddb
Dumping data partition in cfcd0d83f73edfff615ced1aab4f0b3f47847ddb/data_20131222-
2038.dmg ...
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
root@localhost's password:
dd: opening `/dev/rdisk0s2s1': No such file or directory

1799502+0 records in
1799502+0 records out
14741520384 bytes (15 GB) copied, 2797.5 s, 5.3 MB/s
localhost:iphone-dataprotection neda$
```

Методот на физичко извлекување креира нова датотека `data_20131222-2037.dmg`, со dmg екстензија, со име на датотека во која се содржани датумот и времето кога е креирана оваа датотека. Оваа датотека е зачувана во ново креиран директориум чие име го содржи UDID од уредот на iOS кој е предмет на

истражување во овој магистерски труд `cfcd0d83f73edfff615ced1aab4f0b3f47847ddb`.

Копирањето на податоците нема да започне со извршувањето на python скриптата, бидејќи тие се шифрирани и потребно е да отвориме нова терминал командна линија, каде преку извршување на друга скрипта правиме дешифрирање на податоците кои се копираат и автоматски откако ќе бидат дешифрирани, се копираат и се вметнуваат во dmg датотеката. Дешифрирањето на податоците се врши со следнава командна линија:

```
localhost:iphone-dataprotection neda$ python python_scripts/emf_decrypter.py
cfcd0d83f73edfff615ced1aab4f0b3f47847ddb/data_20131222-2038.dmg
Using plist file cfcd0d83f73edfff615ced1aab4f0b3f47847ddb/68cadd73d0700c28.plist
Keybag unlocked with passcode key
cprotect version : 4 (iOS 5)
WARNING ! This tool will modify the hfs image and possibly wreck it if something
goes wrong !
Make sure to backup the image before proceeding
You can use the --nowrite option to do a dry run instead
Press a key to continue or CTRL-C to abort

Decrypting iNode11326
Decrypting iNode11329
Decrypting iNode11336
Decrypting iNode11375
Decrypting iNode11379
Decrypting iNode11380
Decrypting iNode11385
Decrypting iNode11386
Decrypting config.lck
Decrypting config.xml
Decrypting eas.db
Decrypting keyval.db
Decrypting main.db
Decrypting msn.db
Decrypting mediacache.ldb
Decrypting thumbcache.ldb
Decrypting qik_main.db
Decrypting streamlist
Decrypting ecache0
Decrypting 9c97b4c7ce7120cc.dat
Decrypting Cookies.binarycookies
Decrypting 7b6e100c7dffb80d.dat
Decrypting Info.plist
Decrypting KeywordIndex.plist
Decrypting Manifest.sqlitedb
Decrypting express.psa
Decrypted 83424 files
Failed to unwrap keys for :   []
Not encrypted files : 655
```

Откако ќе завршат и двата процеси на дешифрирање и копирање на податоците, ќе бидеме известени со порака во соодветната терминал командна линија, за тоа колку податоци биле декрипирани и колку рекорди и бајти биле ископирани од

уредот. Времетраењето на копирањето зависи од хардверскиот модел на iOS уредот, односно различни хардверски модели значат различни големини на NAND flash меморијата (8 GB, 16 GB, 32 GB, 64 GB).

7. АНАЛИЗИРАЊЕ НА ПОДАТОЦИТЕ ОД ФИЗИЧКАТА КОПИЈА

Често дигиталните уреди се алатка преку која се извршуваат голем број на криминални активности и тие се предмет на форензичка анализа. Овие уреди содржат дигитални докази за извршените криминални активности. Анализирањето на податоците е техника на идентификување, проучување и зачувување на информациите со цел презентација во визуелно разбирлива форма како доказен материјал. Сето ова се прави со цел да се докаже дека биле преземени одредени претходни активности. На пазарот има голем број на мобилни уреди со различен оперативен систем, а поради тоа се разликуваат и техниките кои се применуваат при извлекување и анализирање на податоците. Постои голема разлика при анализирање на дигиталните докази најдени во мобилен уред и во десктоп работна станица, бидејќи мобилните уреди го контролираат пристапувањето кон меморијата, а честопати таа е шифрирана. Не постои универзална форензичка алатка која ќе биде компатибилна речиси со сите мобилни уреди. Дигиталната форензика користи точно дефинирани аналитичко истражувачки техники со цел прибирање докази од дигиталните уреди. Во дигиталната форензика, анализата на извлечените податоци од уредите е главна компонента во целиот процес, и ќе ни помогне да разбереме што и како се случило. Секогаш првото правило во форензиката е да се внимава да не се промени содржината на дигиталниот доказ. Затоа целата работа треба да биде претходно документирана, да се направи форензичка копија и анализите да се прават врз податоците од форензичката копија. Во овој процес не е само да се идентификуваат доказите, потребно е да се најде кој ги креирал, временската рамка како и извлекувањето на избришаните податоци. Често дигиталниот доказ може да биде сокриен, избришан или шифриран, па така целиот процес на анализирање е повеќе од идентификување, прибирање и анализирање. Се применуваат различни техники и алатки за анализирање на податоците и секој најден доказ се документира и се третира исто како физички доказ. Форензичкиот аналитичар потребно е секогаш да знае каде да ги бара доказите и како тие да бидат соодветно толкувани. Голем дел од дигиталните докази се содржани во sqlite и plist датотеки. За анализирање на sqlite

датотеките ќе ја користиме апликацијата SQLiteSpy која е преземена од [24]. За анализирање на plist датотеките ќе ја користиме апликацијата LISTView која е преземена од [25]. За анализирање на геолокациските податоци добиени од мобилната и wi-fi мрежа, ќе ја користиме апликацијата iStalkr која е преземена од [26]. За анализирање и извлекување на геолокациските податоци добиени фотографиите, ќе ја користиме апликацијата Photo GPS Extractor и таа е преземена од [27].

7.1. АНАЛИЗИРАЊЕ НА АДРЕСАРОТ

Адресарот е sqlite база на податоци во која се содржани сите лични контакти и содржи повеќе табели. Содржи табела во која се содржи точното време и датум кога е креиран соодветниот контакт. Сите табели кои содржат датуми во себе, потребно е да се конвертираат од EPOCH временскиот формат. Овој формат е дефиниран како број на секунди кои изминале од 00:00:00 часот, 1 јануари 1970 година. Се наоѓа на следнава локација:
/private/var/mobile/Library/AddressBook/AddressBook.sqlitedb. Нас нè интересираат само неколку поважни табели и тие ќе ги видиме со извршување на следнава sql наредба:

```
select ABPerson.first,ABPerson.last, c.value as MobilePhone, h.value as
HomePhone, datetime(ABPerson.CreationDate+978307200,'unixepoch') as "Creation
Date", datetime(ABPerson.Birthday+978307200,'unixepoch') as "Birthday"
from ABPerson left outer join ABMultiValue c on c.record_id = ABPerson.ROWID
and c.label = 1 and c.property= 3
left outer join ABMultiValue h on h.record_id = ABPerson.ROWID and h.label =
2 and h.property = 3
```

Со извршувањето на наредбата се добива следниов резултат:

First	Last	MobilePhone	HomePhone	Creation Date	Birthday
Doktor goce			75111111	5/29/2013 18:01	7/25/1975 12:34
Ginovce	Park		31111111	5/29/2013 18:01	
balila			023111-111	5/29/2013 18:01	
Filjo			70123456	5/29/2013 18:01	8/26/1987 11:36

Nlb	Tb			5/29/2013 18:01	
Zoran	Zdravev		75123456	5/29/2013 18:01	
Mercedes Mak Auto Star			22580000	5/29/2013 18:01	
Kontakt	Centar		70122	5/29/2013 18:01	
Sp planet			76123333	5/29/2013 18:01	
S			71228326	5/29/2013 18:01	
Idadija t3t4tsh			23113033	5/29/2013 18:01	
Frkac	Biserka		38534526788	5/29/2013 18:01	
Suzana Genel			70123211	5/29/2013 18:01	
Kostarika			23137899	5/29/2013 18:01	
Cabletel			(076) 432-475	5/29/2013 18:01	
Evn	Defekti		89088888	5/29/2013 18:01	
Balkanika			230111111	5/29/2013 18:01	
Cabeltel Kontakt	Smetka		26133613	5/29/2013 18:01	
Olympia	Motors		22600303	5/29/2013 18:01	
Zlatko	Zlatna Raka		+389 70111111	5/29/2013 18:01	
Riversoft	Kompjuteri		23123200	5/29/2013 18:01	
Hr			385989428495	5/29/2013 18:01	
Antena	5		23111911	5/29/2013 18:01	
Faks			47207701	5/29/2013 18:01	
Kolektiva			23109539	5/29/2013 18:01	
Benston	Vesnici		23069564	5/29/2013 18:01	
Panorama	Vrutok Gostivar		70229710	5/29/2013 18:01	
Avtoklima			22771701	5/29/2013 18:01	
Perfeta	Slaykara		23211257	5/29/2013 18:01	

Табела 2. Листа на лични контакти

Tabel 2. Address Book List

7.2. АНАЛИЗИРАЊЕ НА ТЕЛЕФОНСКАТА КОМУНИКАЦИЈА

Целата телефонска комуникација е документирана во call_history.db датотека која е sqlite база на податоци и се наоѓа во: /private/var/wireless/Library/CallHistory/call_history.db. Содржи повеќе табели од кои најбитната табела е call табелата. Во оваа табела има поле flags кое содржи неколку вредности и тоа: доколку вредноста на полето е 4 станува збор за

дојдовен повик, доколку вредноста на полето е 5 станува збор за појдовен повик и доколку вредноста на полето е 8 станува збор за блокиран повик. Со извршување на следнава sql команда, ќе ја добиеме телефонската комуникација табела 3:

```
select datetime(date,'unixepoch','localtime') as 'Date', case flags when '4'
then 'Incoming' when '5' then 'Outgoing' when '8' then 'Blocked' else
'Unknown' end as 'Call Type(from Flags)', address as 'Phone #', duration as
'duration(seconds)' from call
```

Date	Call Type(from Flags)	Phone #	duration(seconds)
11/23/2013 11:43	Unknown	70123456	0
11/23/2013 14:26	Incoming	70123457	19
11/24/2013 10:59	Incoming	70123458	167
11/24/2013 14:08	Incoming	70123459	35
11/24/2013 14:14	Outgoing	70123460	20
11/24/2013 15:39	Incoming	70123461	32
11/24/2013 16:16	Incoming	70123462	515
11/24/2013 16:30	Outgoing	70123463	0
11/24/2013 16:30	Outgoing	70123464	98
11/25/2013 7:57	Outgoing	70123465	53
11/25/2013 9:01	Incoming	70123466	170
11/25/2013 10:07	Outgoing	70123467	23
11/25/2013 10:31	Blocked		0
11/25/2013 10:38	Outgoing	71123489	29
11/25/2013 10:53	Outgoing	71123490	0
11/25/2013 10:55	Incoming	71123491	22

Табела 4. Листа на телефонската комуникација

Tabel 4. Telephone communication list

7.3. АНАЛИЗИРАЊЕ НА ТЕКСТУАЛНИТЕ ПОРАКИ

Базата на податоци на текстуалните пораки е единствената база на податоци која во себе ги содржи како пратените и добиените текстуални пораки така и избришаните текстуални пораки. Sms.db е sqlite база на податоци и таа се наоѓа во /private/var/mobile/Library/SMS. Содржи повеќе табели од кои најбитната табела е message табелата. Во оваа табела има поле flags кое содржи неколку вредности

и тоа: доколку вредноста на полето е 2 станува збор за примена порака, доколку вредноста на полето е 3 станува збор за испратена порака, доколку вредноста на полето е 33 станува збор за неуспешно испратена порака и доколку вредноста на полето е 129 станува збор за избришана порака. Со извршување на следнава sql команда, ќе ги добиеме сите текстуални пораки:

```
Select address as "Phone Number", text as "Message", datetime (message.date,
'unixepoch', '+1 hours') as "Date Read/Sent",
CASE flags
WHEN 2 THEN "Received"
WHEN 3 THEN "Send"
WHEN 33 THEN "Failure"
WHEN 129 THEN "Deleted"
ELSE "Unknown"
END as Type
from message
```

Резултатот е следнава табела:

Phone Number	Message	Date Read/Sent	Type
71123456	Me fat festa e bajramit, ju uroj shendet dhe suksese ne jete. Sashe	8/8/2013 17:26	Failure
T-Mobile	T-Mobile vi posakuva srekjen pat! Za poevtini razgovori kon Makedonija birajte *123*07XXXXXXX# Za pomosh birajte *123*070122#	12/6/2010 7:03	Received
T-Mobile	Dokolku koristite mobilen internet napomenuvame deka cenata vo stranstvo e povisoka od doma. Povekje informacii na *123*070122#	12/6/2010 7:03	Received
144144	A4 SK435RJ	12/13/2010 15:30	Send
144144	Parkiranjeto na voziloto SK435RJ e dozvoleno do 13.12.10 17:30, potoa treba da go otstranite. Za kraj, ispratete “S” na istiot broj.	12/13/2010 15:29	Received
144144	Parkiranjeto na voziloto SK0501AB e zapocnato. Tarifata vazi do zavrsuvanje na vremeto na parkiranje vo 20.12.13 21:00. Za kraj ispratete “S” na 144144.	12/20/2013 12:24	Received
144144	S	12/20/2013 12:46	Send
144144	Parkiranjeto zavrsi vo 20.12.13 12:47. Vkupen nadomestok 25,00 MKD. Pri povtorno parkiranje so ist mobilen i vozilo ispratete ja samo zonata na 144144.	12/20/2013 12:47	Received
T-Mobile	Imate propushteni povici od 071380932 (1) 21/12/13 12:03	12/21/2013 12:03	Deleted

Табела 5. Листа на текстуални пораки

Tabel 5. List of SMS

7.4. АНАЛИЗА НА ПРЕБАРУВАНИ ЗБОРОВИ

Сите клучни зборови кои биле пребарувани преку вградениот Safari интернет-пребарувач зачувани во XML формат во датотеката RecentSearches.plist, лоцирана во /private/var/mobile/Library/Caches/Safari.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>RecentSearches</key>
	<array>
		<string>evropa gumi</string>
		<string>cat6</string>
		<string>wifi psk</string>
		<string>bambisimo kapistec</string>
		<string>16 челични бандаши</string>
		<string>дрвена ламперија за таван</string>
		<string>дрвена ламперија</string>
		<string>дозвола за управување на туѓо возило од сопругата</string>
		<string>дозвола за управување на туѓо возило</string>
		<string>phone numbering</string>
		<string>nlbklik mk</string>
		<string>dd command divide output file</string>
		<string>flights tirana to new york</string>
		<string>tirana airport</string>
		<string>bmw 750 12 cylinder</string>
		<string>Mlin balkan idadija</string>
		<string>Mlin balkan</string>
		<string>china post tracking</string>
		<string>iphone backup user partition ssh ramdisk</string>
		<string>iphone backup user partition</string>
	</array>
</dict>
</plist>
```

7.5. АНАЛИЗИРАЊЕ НА dynamic-text.dat

На iOS уредите, сè што е внесено преку тастатурата на уредот, преку која било апликација е зачувано во датотеката dynamic-text.dat со локација во /private/var/mobile/Library/Keyboard/. Кој било збор внесен во апликации како Notes, Messages, Facebook и др., кои дозволуваат внесување текст, ќе бидат зачувани

[13]. Содржината на оваа датотека може да се види доколку таа се отвори со која било апликација за уредување текст. Внатрешноста на оваа датотека изгледа вака:

```
DynamicDictionary-
4____tamu_ima_pat_api_not_gmail_iphone_home_new_casio_ima_protect_aga
inst_sql_forensics_iphone_facebook_motion_iphone_facebook_crashes_airp
rint_airprint_air_print_airprint_gmail_pro_printer_digital_repair_sony
_pro_Ringtones_gmail_gmail_site_CYPRESS_bottom_print_renault_sport_way
_voice_room_temperature_thermometer
paypal_gmail_skype_ahmed_terrorist_______
     _aah__ab__ac__accu__acharavi__Acousticruler__adresa__adsheet__ae
rodrom__agencija__agia__ahowing__aixam__aj__ajde__akcijata_!godini__go
dishni__golema__GOLEMI__golemo__gori__gorica__gorivo____Gostivar__gost
ive__Gotie__gotov__gotova__gotovi__grabo__gradezni__gradot__gradski__g
rama_
```

7.6. АНАЛИЗИРАЊЕ НА БЕЛЕШКИТЕ

Сите белешки внесени од страна на уредот се зачувани во sqlite датотека notes.sqlite, лоцирана во /private/var/mobile/Library/Notes/. Со извршување на следнава sql команда, ќе ги видиме сите белешки:

```
select ZTITLE AS TEXT, ZSUMMARY AS TEXT1,
datetime(ZCREATIONDATE+978307200,'unixepoch','localtime') AS DATE FROM ZNOTE
```

Резултатот од sql командата е следниов:

TEXT	TEXT1	DATE
Педесет нијанси на Греј (сиво) - Е. Л. Џејм		12/26/2012 22:01
Iphone 4s a1387 bela boja		1/3/2013 11:33
Katerinazoran		8/11/2013 12:43
Autocad 2010	Photoshop Maya Solidworks	10/27/2013 13:16
Shutdown -rS now		10/31/2013 14:46
0B96B9F4-03BA-4774-B912-508151B10C3E		11/29/2013 14:06

Табела 6. Преглед на белешките

Tabel 6. Overview of the notes

7.7. АНАЛИЗИРАЊЕ НА КАЛЕНДАРОТ

Во уредите на iOS има опција за поставување потсетник за идни закажани состаноци, родендени или некој друг поважен датум. Доколку преку уредот се најавиме на facebook или на gmail, вградената апликација calendar автоматски ќе ги преземе сите потсетници со датум и со време доколку тие постојат. Датотеката Calendar.sqlite се наоѓа во /private/var/mobile/Library/Calendar/. Со извршување на следнава sql команда, ќе ја видиме содржината на sql базата на податоци:

```
select summary, datetime(start_date+978307200, 'unixepoch') as Birthday from
CalendarItem
```

Резултатот од sql командата е табела 7.

summary	Birthday
Denis reno's Birthday	13/12/2008 0:00
Matea's Birthday	12/1/2004 0:00
Igorka's Birthday	21/7/1979 0:00
Marija Pop-Petrovska Pop-Petrovska Gjeorgievska's Birthday	1/11/1979 0:00
Sostanok	11/11/2013 09:45
Anhoch HDD 1TB	29/12/2013 12:30
Filjo's Birthday	26/8/1987 0:00
Andreja's Birthday	30/11/1978 0:00

Табела 7. Податоци во календар базата на податоци

Tabel 7. Calendar database data

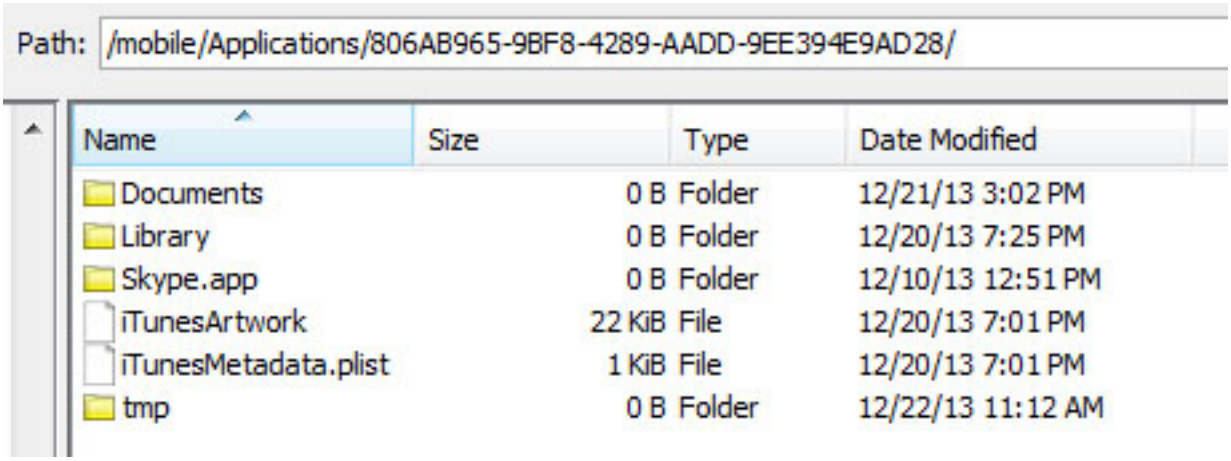
7.8. АНАЛИЗИРАЊЕ НА АПЛИКАЦИИТЕ

Сите апликации инсталирани на iPhone, преку iTunes, се креира директориум во private/var/mobile/Applications. Името на директориумот е единствен идентификациски стринг кој содржи 32 алфанумерички знаци (0B96B9F4-03BA-4774-B912-508151B10C3E). Овој стринг е различен за секоја нова апликација. Структурата на овој директориум е следнава:

- Documents – се користи за зачувување кориснички документи и податоци за апликацијата во sqlite database формат;

- Library – содржи неколку поддиректориуми и овде не се зачувуваат кориснички податоци. Се користи за зачувување специфични апликациски датотеки:
    - Library/Preferences – содржи датотеки со својства за самата апликација;
    - Library/Caches – содри податоци симнати преку интернет од самата апликација. Податоците во овој директориум ги брише самата апликација по потреба.
- Tmp – се зачувуваат привремени датотеки.

Во овие директориуми можеме да најдеме многу податоци, како што се приспособувања за апликацијата, претходни активности, кориснички имиња, лозинки и др.



Слика 11. Хиерархиска структура

Figure 11. Hierarchial structure

7.9. АНАЛИЗИРАЊЕ НА И-МЕЈЛ ПОДАТОЦИ

Доколку на уредите на iOS е конфигурирано користењето на и-мејл, тогаш сите конфигурациски датотеки и целата комуникација на и-мејл е зачувана во /private/var/mobile/Library/Mail. На оваа локација можеме дури и да ги видиме сите

прикачени датотеки за време на комуникацијата на и-мејл. Датотеката metadata.plist, ги содржи сите конфигурирани и-мејл кориснички сандачиња и тоа:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>FetchingData</key>
	<dict>
		<key>as://saso.21072@student.ugd.edu.mk/A33D4E68-0DF0-4647-A9E5-8C2040B21282/5</key>
		<date>2013-12-18T16:36:48Z</date>

	<key>imap://sasogmk%40gmail.com@imap.gmail.com/%5BGmail%5D/%5CDrafts</key>
		<date>2013-04-04T16:26:54Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/%5BGmail%5D/%5CSent</key>
		<date>2013-10-17T08:16:33Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/%5BGmail%5D/%5CSpam</key>
		<date>2013-12-08T06:18:36Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/%5BGmail%5D/%5CTrash</key>
		<date>2013-07-11T10:40:50Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/CRYO-SAVE</key>
		<date>2013-04-24T20:44:44Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/INBOX</key>
		<date>2013-12-18T16:36:50Z</date>
		<key>imap://sasogmk%40gmail.com@imap.gmail.com/Sent%20Items</key>
		<date>2013-04-04T07:26:46Z</date>
		<key>imap://sasogmk%40yahoo.com@apple.imap.mail.yahoo.com/Bulk%20Mail</key>
		<date>2013-10-31T16:02:34Z</date>
		<key>imap://sasogmk%40yahoo.com@apple.imap.mail.yahoo.com/INBOX</key>
		<date>2013-12-17T16:02:50Z</date>
		<key>imap://sasogmk%40yahoo.com@apple.imap.mail.yahoo.com/Sent</key>
		<date>2013-10-31T16:03:39Z</date>
		<key>local:///Outbox</key>
		<date>2013-09-17T11:55:09Z</date>
	</dict>
	<key>OverflowData</key>
	<dict>
		<key>LastLoadOlder</key>
		<date>2013-07-09T10:05:37Z</date>
	</dict>
	<key>SearchData</key>
	<dict>
		<key>as://saso.21072@student.ugd.edu.mk/4767D317-4B10-414F-86DC-D1DBC24DCC6F/5</key>
		<date>2013-05-20T07:46:30Z</date>
```

```
                <key>as://saso.21072@student.ugd.edu.mk/A33D4E68-0DF0-4647-A9E5-
8C2040B21282/5</key>
                <date>2013-10-08T12:05:07Z</date>
                <key>imap://sasogmk%40gmail.com@imap.gmail.com/INBOX</key>
                <date>2013-12-05T10:09:49Z</date>
                <key>imap://sasogmk%40yahoo.com@apple.imap.mail.yahoo.com/INBOX</key>
                <date>2013-11-06T13:30:34Z</date>
        </dict>
</dict>
</plist>
```

Целата комуникација на и-мејл е зачувана во sqlite базата на податоци Protected Index. Иако оваа датотека нема никаква препознатлива екстензија дека е sqlite база на податоци, сепак, со извршување на следнава sql команда ќе ја видиме содржаната комуникација со и-мејл:

```
SELECT * FROM "messages";
```

Резултатот е табела 8

message_id	sender	subject	_to	cc	bcc
1422	eBay	6 days only—collectible sneakers.	sasogmk@yahoo.com		
1423	"Facebook"	Баже Огнаноски updated his status: "Mirisot na peceni krilca..."	Sinisha Gjeorgievski		
1441	eBay	Collectible sneaker event ends soon!	sasogmk@yahoo.com		
1464	"Facebook"	Do you know Mila Angeleska, Magi Magdalena and 6 others?	Sinisha Gjeorgievski		
1893	"UGD Contact"	концерт на Универзитетски хор	"UGD Everyone"	"UGD Studenti 1 Ciklus" "UGD Studenti 2 Ciklus"	
1910	"Facebook"	Marija Pop-Petrovska Gjeorgievska, Баже Огнаноски and 36 others are following Pages on Facebook	Sinisha Gjeorgievski		
6542	TechRepublic Daily Digest	10 ways to survive a critical system outage	sasogmk@gmail.com		
6543	"KONIKO Sales"	IZDVOJUVAME : NOVI KOLICINI NA CHIEFTEC NAPOJUVANJA OD 400W do 1.000W	"Mailing List Members"		

Табела 8. Комуникација на и-мејл

Tabel 8. Email comunication

7.10. АНАЛИЗИРАЊЕ НА ГОВОРНОТО САНДАЧЕ

Сите говорни пораки кои се преслушани се зачувани на локација /private/var/mobile/Library/Voicemail/ со amr екстензија. Во истиот директориум се наоѓа и база на податоци со сите информации за секоја говорна порака како: време и датум кога е оставена говорната порака, од кој телефонски број.

```
== .Voicemails$Voicemail Entries ==
= Entry =
[
        CallbackNumber = null,
        FilenameNumber = 5,
        DateDate = 8/21/13 3:11 PM,
        Sender = +38971111111,
        = Entry =
[
        CallbackNumber = null,
        FilenameNumber = 6,
        DateDate = 8/25/13 7:19 PM,
        Sender = +38976111111,
```

Доколку аудио говорните пораки се во Voicemail директориумот, тие можеме да ги преслушаме со користење соодветна аудио апликација како QuickTime. Во Voicemail директориумот се наоѓа com.apple.voicemail.imap.parameters.plist датотека во која се содржат информации за говорното сандаче:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
        <key>AccountSettings</key>
        <dict>
                <key>name</key>
                <string>70365844@mejli.mobimak.com.mk</string>
                <key>port</key>
                <string>143</string>
                <key>server</key>
                <string>62.162.129.55</string>
        </dict>
        <key>AccountState</key>
        <string>Active</string>
        <key>BeaconCount</key>
        <integer>0</integer>
        <key>GreetingType</key>
        <string>Standard</string>
        <key>SyncRequired</key>
        <false/>
```

```
</dict>
</plist>
```

7.11. АНАЛИЗИРАЊЕ МРЕЖНИ АРТЕФАКТИ

Сите мрежни информации поврзани со овој уред се наоѓаат во повеќе датотеки на различни локации и содржат различни видови информации:

- Доколку уредот на iOS бил конфигуриран во функција да го дели мобилниот интернет со други уреди преку безжичната мрежа, тогаш сите уреди кои ја користеле оваа услуга преку овој уред на iOS, ќе бидат евидентирани во dhcpd_leases датотека со локација во /private/var/db. Содржината на оваа датотека можеме да ја видиме со кој било текстуален уредувач и таа изгледа вака:

```
{
      name=Lenovo-PC
      ip_address=172.20.10.8
      hw_address=1,de:2b:61:19:a4:5
      identifier=1,de:2b:61:19:a4:5
      lease=0x5242c118
}
{
      name=Vaio-VAIO
      ip_address=172.20.10.5
      hw_address=1,78:dd:8:fa:2e:2a
      identifier=1,78:dd:8:fa:2e:2a
      lease=0x51c95ea5
}
```

За секој уред кој што ќе се поврзе со уредот на iOS и ќе ја користи услугата наречена hotspot, се запишуваат во ASCII датотека името на уредот, ip адресата која му се доделува и MAC адресата на неговата мрежна картица со која пристапува кон оваа услуга.

- Секогаш кога уредите на iOS ќе се најават на безжична интернет-мрежа, во специјална датотека се запишуваат податоци за мрежата во */private/var/preferences/SystemConfiguration*. Податоците се запишани во

com.apple.wifi.plist датотека и најбитните податоци се: името на мрежата која зрачи, типот на автентикацискиот процес, датум кога последен пат се пријавил на мрежата и хардверската MAC адреса на уредот. Дел од содржината на оваа датотека изгледа вака:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<key>SSID</key>
                        <data>
                        TUVESVRFUlJBTkVBTl9DT1NNT1M=
                        </data>
                        <key>SSID_STR</key>
                        <string>MEDITERRANEAN_COSMOS</string>
                        <key>ScaledRSSI</key>
                        <real>0.60375714302062988</real>
                        <key>ScaledRate</key>
                        <real>1</real>
                        <key>Strength</key>
                        <real>0.60375714302062988</real>
                        <key>WEPKeyLen</key>
                        <integer>0</integer>
                        <key>authMode</key>
                        <integer>0</integer>
                        <key>enabled</key>
                        <true/>
                        <key>isValid</key>
                        <true/>
                        <key>isWPA</key>
                        <integer>0</integer>
                        <key>lastAutoJoined</key>
                        <date>2013-10-26T08:51:48Z</date>
                        <key>lastJoined</key>
                        <date>2013-09-10T11:15:11Z</date>
                        <key>networkChannelListKey</key>
                        <dict>
                                <key>1</key>
                                <dict>
                                        <key>BSSID</key>
                                        <string>0:3a:99:7c:d8:60</string>
                                        <key>CHANNEL</key>
                                        <integer>1</integer>
                                        <key>CHANNEL_FLAGS</key>
                                        <integer>8</integer>
                                </dict>
```

7.12. АНАЛИЗИРАЊЕ ГЕОЛОКАЦИСКИ ПОДАТОЦИ

Мобилните уреди на iOS се со GPS функционалност и секогаш кога ќе се побара одредена локација или одредување на моменталната локација, податоците се зачувуваат во соодветни датотеки. Има повеќе локации каде се складираат овие податоци, и тоа:

- Мобилните уреди на iOS со вградена GPS функционалност, не секогаш добиваат добар GPS сигнал. Секогаш пред да се побара GPS сигнал, се добива груба проценка за моменталната локација преку мобилната мрежа од најблиските базни станици. Доколку е неуспешно добивањето на точната локација и лоцирањето на GPS сателитите, уредот се обидува преку wi-fi мрежата да ја добие точната локација. Овие wi-fi јазли можат да помогнат при одредувањето на точната моментална локација. Сите овие податоци, дали се добиени преку мобилната мрежа или преку wi-fi мрежата, се складирани во различни табли во consolidated.db. Во оваа sqlite база на податоци, податоците се зачувуваат само кога одредена апликација, која ги користи GPS сервисите, ќе ја побара моменталната локација. Оваа база на податоци се наоѓа на локација /root/Library/Caches/locationd/. Оваа sqlite датотека ќе ја отвориме со апликацијата iStalkr. Оваа апликација знае да ги одвои податоците добиени од мобилната и wi-fi мрежа и тие се експортираат во датотека со формат компатибилен за преглед во Google Earth. Табелата со податоци добиени од мобилната мрежа изгледа вака:

MCC	MNC	LAC	CI	Timestamp	Latitude	Longitude
294	2	21100	34332	409417778.8	42.0255383	21.44343641
294	2	21100	30311	409417778.8	42.0255383	21.44072172
294	2	21100	4332	409417778.8	42.0255383	21.44099812
294	2	21100	3821	409417778.8	42.03545317	21.43830318
294	3	3100	-1	409417781.6	42.14959637	21.87244015

294	1	111	-1	409417781.6	42.1392915	21.87966872
212	1	101	-1	409417781.6	42.42456657	21.14352075
294	1	210	2049	409058782.4	41.99849774	21.4243467
294	1	210	728321	409058782.4	41.99849774	21.42482844
294	1	210	2067	409058782.4	41.99779061	21.42530767
294	3	2002	116913	409058782.4	41.99849774	21.4338116
294	3	2002	116423	409058782.4	41.99825689	21.43369158
294	1	210	2105	409424223.5	42.00520304	21.35156581
294	2	21100	99459	409424223.5	42.00883375	21.36160983
294	3	2002	115553	409424223.5	42.00751126	21.36160048
294	1	210	728317	409058782.4	41.98944663	21.42389172

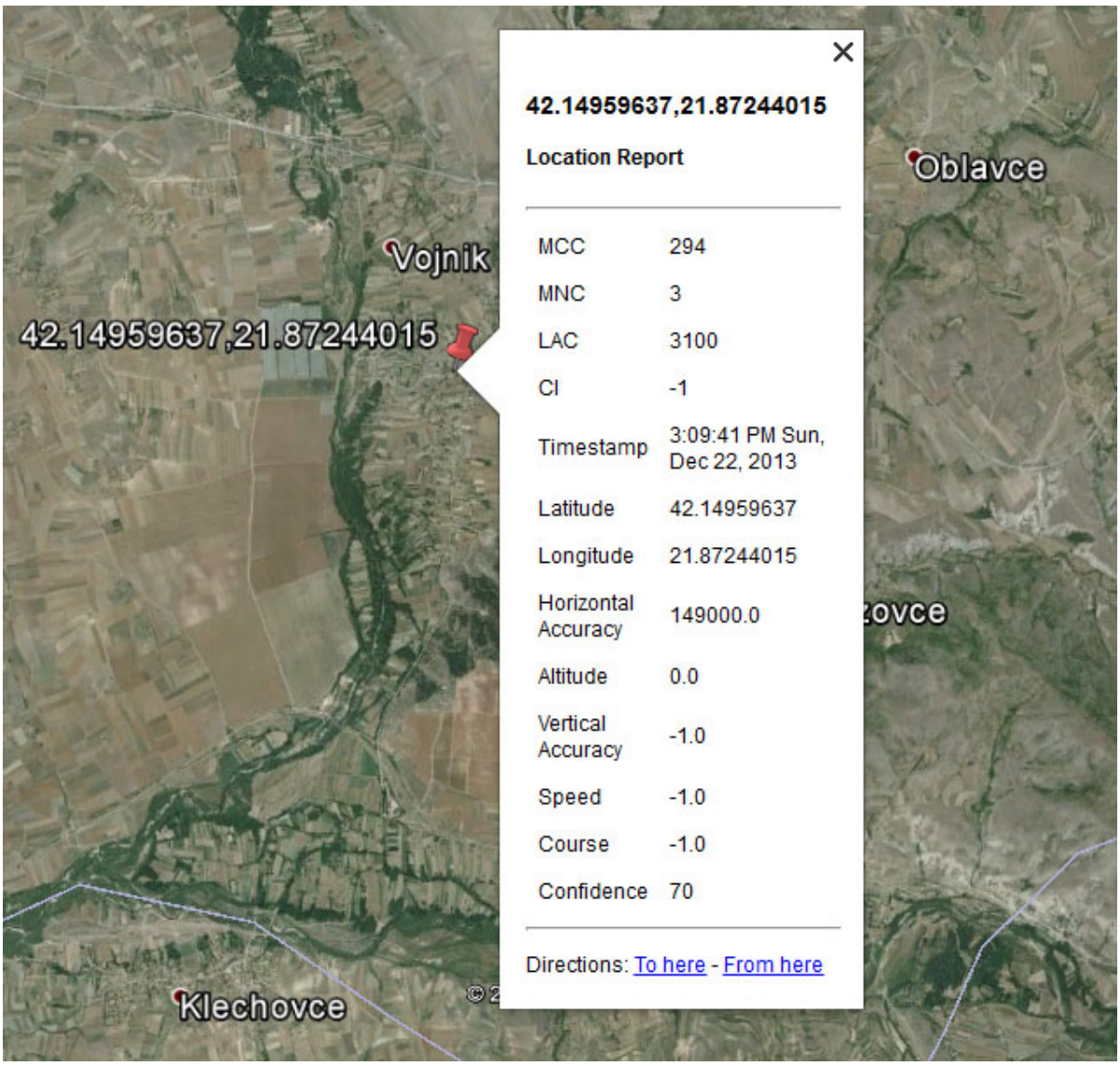
Табела 9. Геолокациски податоци од мобилната мрежа

Tabel 9. Geolocation data from the cellular network

MCC колоната е код кој соодветствува на државата во која е добиена точната локација преку мобилната мрежа. Во табелата 9, кодот 294 соодветствува за државата Македонија, а 212 за државата Косово. MNC колоната е код кој соодветствува на соодветниот мобилен оператор преку кој е добиена точната локација. Па така за MCC 294 (Македонија), следниве кодови за MNC се:

- 1 е T-Mobile;
- 2 е One;
- 3 е VIP.

Колоната со време го претставува времето кога мобилниот уред iPhone бил на соодветната локација. Може да се забележи дека истата вредност на timestamp колоната е зачувана за неколку внесени различни геолокации. Тоа значи дека во исто време на моменталната локација на која бил iPhone, со цел да се направи триангулација, зрачат повеќе базни станици означени во табелата CI (Cell Id). Доколку истите податоци ги експортираме во kml формат, тие ќе можеме визуелно да ги видиме во апликација за приказ на информации со географски мапи, а резултатот би бил следниов:



Слика 12. Геолокациските податоци прикажани во Google Earth

Figure 12. Geolocation data shown in Google Earth

- Податоците добиени преку wi-fi мрежите се зачувани во посебна табела и содржината изгледа вака:

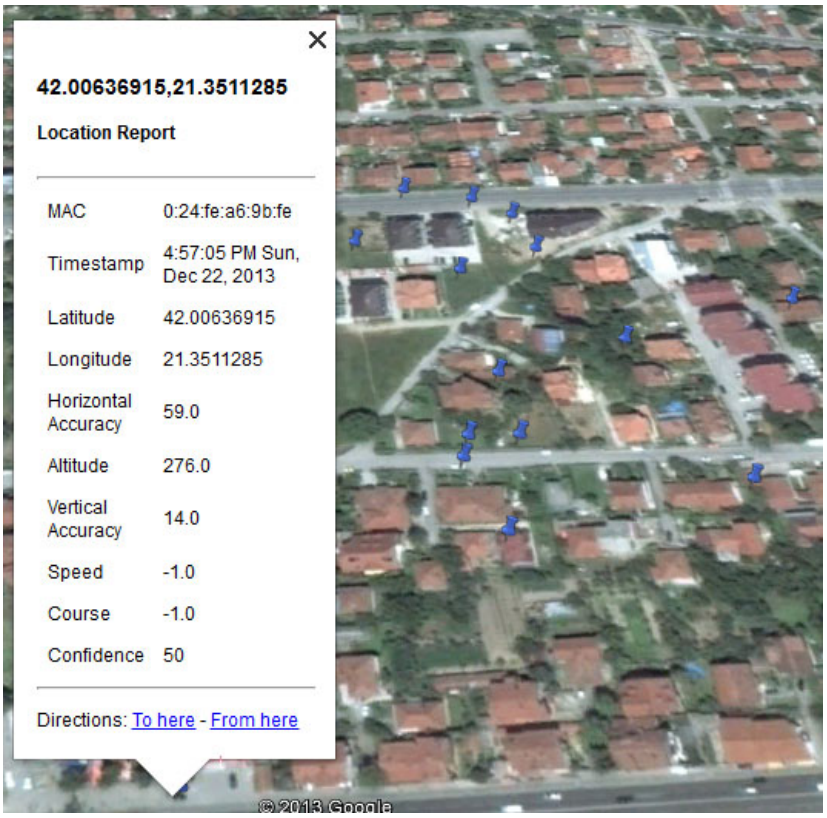
MAC	Channel	Timestamp	Latitude	Longitude
34:e0:cf:6c:1a:52	11	22/12/2013 16:57	42.00750563	21.352194
0:21:29:b6:2:2	11	22/12/2013 16:57	41.99922289	21.40530336
74:ea:3a:eb:b8:8e	1	22/12/2013 16:57	42.00807323	21.34774433
34:e0:cf:6c:8a:46	1	22/12/2013 16:57	42.00779795	21.35150404

90:f6:52:b5:a5:dc	4	22/12/2013 16:57	42.00856474	21.35176349
f8:d1:11:af:8:94	1	22/12/2013 16:57	42.00651225	21.35367743
0:25:9c:31:78:3e	6	22/12/2013 16:57	42.00853415	21.35262897
a0:f3:c1:65:18:d6	6	22/12/2013 16:57	42.00744081	21.35338071
90:f6:52:25:37:94	6	22/12/2013 16:57	42.00759826	21.35222243
0:24:fe:8f:4:ef	1	22/12/2013 16:57	42.00826377	21.35377839

Табела 10. Геолокациски податоци од wi-fi мрежа

Tabel 10. Geolocation data from the wi-fi network

Доколку истите податоци ги експортираме во kml формат, тие ќе можеме визуелно да ги видиме во апликација за приказ на информации со географски мапи, а резултатот би бил следниов:



Слика 13. Геолокациските податоци прикажани во Google Earth

Figure 13. Geolocation data shown in Google Earth

Секогаш кога ќе побараме да добиеме информација за моменталната локација се собираат и зачувуваат податоци за блиските мобилни базни станици и wi-fi

пристапни точки. Истите податоци се запишуваат доколку активираме апликација која бара пристап до GPS функционалноста. Доколку геолокациските сервиси се вклучени, iPhone ги праќа на Apple на секои 12 часа геолокациските податоци анонимно со случаен идентификациски број.

- Доколку е конфигурирано при користењето на вградената камера на мобилните уреди на iOS, при секое фотографирање, фотографијата се зачувува на соодветна локација, но, исто така, во EXIF[15] податоците се зачувуваат и GPS координатите од локацијата каде моменталната фотографија е фотографирана. Откако сите директориуми кои ги содржат фотографиите, ќе бидат скенирани од страна на апликацијата Photo GPS Extractor, имаме опција резултатот од скенирањето да биде зачуван како во табела 11 или во kml формат за приказ на географска мапа.

G:\Data\mobile\Media\DCIM\101APPLE\IMG_1264.JPG	42.0145	21.391	273
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1265.JPG	42.0145	21.39067	280
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1266.JPG	41.9805	21.47217	280
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1267.JPG	41.98033	21.47567	268
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1270.JPG	42.00783	21.35183	195
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1271.JPG	42.00783	21.35183	218
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1280.JPG	42.00767	21.35233	281
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1281.JPG	42.00767	21.3525	280
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1282.JPG	42.00783	21.35233	274
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1283.JPG	42.00783	21.35233	274
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1284.JPG	42.00783	21.35233	274
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1285.JPG	42.00783	21.35233	277
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1286.JPG	42.01467	21.38783	245
G:\Data\mobile\Media\DCIM\101APPLE\IMG_1287.JPG	42.00433	21.39067	280

Табела 11. Геолокациски податоци содржани во фотографии

Tabel 11. Geolocation data contained in photos

- Доколку преку вградената апликација за геонавигирање – maps, сме пребарале некоја локација по име на град или на улица, ќе бидат зачувани во /private/var/mobile/Library/Maps/history.plist, чија содржина е следнава:

```
<?xml version="1.0" encoding="UTF-8"?>
```

[15] Стандард кој го специфицира стандардот на сликите.

```
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
	<key>HistoryItems</key>
	<array>
		<dict>
			<key>DisplayQuery</key>
			<string>Zakintos</string>
			<key>HasMultipleLocations</key>
			<false/>
			<key>HistoryItemType</key>
			<integer>0</integer>
			<key>Latitude</key>
			<real>41.993244171142578</real>
			<key>LatitudeSpan</key>
			<real>0.011865200474858284</real>
			<key>Location</key>
			<string>Zakynthos</string>
			<key>Longitude</key>
			<real>21.421194076538086</real>
			<key>LongitudeSpan</key>
			<real>0.01373291015625</real>
			<key>Query</key>
			<string>Zakintos</string>
			<key>SearchKind</key>
			<integer>2</integer>
			<key>ZoomLevel</key>
			<integer>16</integer>
		</dict>
		<dict>
<dict>
			<key>DisplayQuery</key>
			<string>Vodnjanska</string>
			<key>HasMultipleLocations</key>
			<false/>
			<key>HistoryItemType</key>
			<integer>0</integer>
			<key>Latitude</key>
			<real>13.795406341552734</real>
			<key>LatitudeSpan</key>
			<real>3.9682748317718506</real>
			<key>Location</key>
			<string>Zagreb</string>
			<key>Longitude</key>
			<real>-88.9013671875</real>
			<key>LongitudeSpan</key>
			<real>3.515625</real>
			<key>Query</key>
			<string>Vodnjanska</string>
			<key>SearchKind</key>
			<integer>2</integer>
			<key>ZoomLevel</key>
			<integer>8</integer>
		</dict>
```

7.13. АНАЛИЗИРАЊЕ НА keychain-2.db

Во уредите на iOS, keycahin е sqlite база на податоци со локација /private/var/Keychains. Оваа база на податоци содржи многу кориснички податоци, како кориснички имиња, лозинки и др. кои биле користени на уредот [13]. Понекогаш лозинката е шифрирана и потребно е да се дешифрира. Оваа база на податоци е шифрирана со хардверскиот клуч кој е единствен за секој уред. Со користење на Python скрипта врз keychain.db, резултатот е следен:

Service	Account	Data	Access group	Protection class
iTools	sasogmk	anakonda	apple	Always
AirPort	WLAN-0024FE861989	slovachka	apple	AfterFirstUnlock
AirPort	Pro-Beting	slovacka	apple	AfterFirstUnlock
MobileBluetooth	00:05:4F:69:76:2D	<?xml version="1.0"	apple	AlwaysThisDeviceOnly
MobileBluetooth	00:10:48:E8:70:FD	<?xml version="1.0"	apple	AlwaysThisDeviceOnly
AirPort	GizziGrill.com	123456789	apple	AfterFirstUnlock
twitter			J8J28689H5.com.ebay.ebaydeals	WhenUnlocked
token	token	1	U6C235RAZ4.de.iosphere.offmaps2	WhenUnlocked
loyalFlatRateUser	loyalFlatRateUser	FALSE	U6C235RAZ4.de.iosphere.offmaps2	WhenUnlocked
AirPort	APOLLONHOTEL	apollonhotel	apple	AfterFirstUnlock
AirPort	tassos	villatassos	apple	AfterFirstUnlock
	ComputerCredentials:sasogmk@gmail.com:74	>0200000006000000280	3P33PBCE25.com.logmein.ignition	WhenUnlocked
AccessCode	ComputerCredentials:sasogmk@gmail.com:74	>0200000006000000280	3P33PBCE25.com.logmein.ignition	WhenUnlocked
	sasogmk@gmail.com	Anakonda31	X56N753KSK.com.mathworks.matlab	WhenUnlocked
AirPort	Almira 2nd floor	BB82079B6A	apple	AfterFirstUnlock
AirPort	_AppleWi-FiInternetTetheringSSID_	tubby3392	apple	AfterFirstUnlock
https://www.nlbklik.com.mk	1saso.georgievski	123456789	6VPQ6KBZ53.com.dolphin.browser.iphone	WhenUnlocked
AirPort	Cisco	nedameda18102012	apple	AfterFirstUnlock
AirPort	Speedport wifi	g00gl32011mkd	apple	AfterFirstUnlock
AirPort	SAMMYS2	BE99155719	apple	AfterFirstUnlock
AirPort	SAMMYS3	BE99155719	apple	AfterFirstUnlock
com.viber	viberDeviceKey	6bb756bc89f1c455d7c9	69V327AA4Z.com.viber	WhenUnlocked
AirPort	GYROS ARISTOTELOUS	bestplace	apple	AfterFirstUnlock
iAliexpress	userName	sasogmk@gmail.com	QYA43G95L4.com.alibaba.iAliexpress	WhenUnlocked
iAliexpress	loginId	mk1042407566	QYA43G95L4.com.alibaba.iAliexpress	WhenUnlocked
iAliexpress	memberSeq	162502804	QYA43G95L4.com.alibaba.iAliexpress	WhenUnlocked
iAliexpress	atmToken	y2ap+XQC2rCZ3tkyGXND	QYA43G95L4.com.alibaba.iAliexpress	WhenUnlocked

Табела 12. Содржината на keychain.db

Tabel 12. Contents of keychain.db

8. СЛИЧНИ КОРИСНИЧКИ БАЗИ НА ПОДАТОЦИ КАЈ УРЕДИТЕ НА ANDROID

Постојат многу сличности и само неколку разлики помеѓу оперативните системи iOS и Android. Кај оперативниот систем Android се користат XML датотеки за зачувување на системските конфигурациски параметри. Понекогаш овие податоци се зачувани во SQLite бази на податоци. Најголемата разлика е тоа што кај уредите на Android има можност за опцијата ADB Shell, која овозможува пристап до податочниот систем и извлекување на податоците за анализа. Количината на податоците што може да се извлече зависи од тоа дали уредот е рутиран или не. За да имаме комплетен пристап до сите податоци на мобилниот уред на Android, односно пристап до root директориумот, потребно е уредот да е рутиран (процес сличен на jailbreak кај уредите на iOS). Android повеќето од податоците ги зачувува во root директориумот (/) со root кориснички привилегии.

Повеќето уреди на Android се партиционирани на следниов начин [28]:

- /boot – ги содржи ramdisk и кернелот, кои овозможуваат уредот да се стартува;
- /system – оваа партиција го содржи оперативниот систем со вградените апликации;
- /recovery – е втора boot партиција која овозможува стартување на уредот во recovery мод со цел да се отстранат грешките при стартување на уредот или да се обнови моменталната инсталација на оперативниот систем;
- /data – ги содржи корисничките податоци како телефонскиот адресар, историја на телефонски повици, податоци од апликации од трети страни инсталирани од страна на корисникот;

- /cache – со користење на уредот, во оваа партиција се зачувуваат како информации најчесто користените податоци;
- /misc – содржи системски приспособувања, како и информации за хардверот.

Апликациските податоци се зачувани во /data партицијата со локација /data/data/име на апликацијата/databases. Освен во /data партицијата, понекогаш има и привремени податоци во /cache партицијата. Понекогаш податоците се сместени и на NAND flash меморијата и на надворешната SD мемориска картица.

- Како кај уредите на iOS така и уредите на Android, ги зачувуваат во SQLite база на податоци сите зборови внесени преку стандардната вградена тастатурата. Локацијата на оваа датотека е:

  /data/data/com.android.providers.userdictionary/databases/user_dict.db. Доколку се користи тастатура инсталирана преку трети апликации од страна на корисникот, зборовите нема да бидат зачувани во оваа база на податоци;

- ContextDB.db – база на податоци само за мобилните уреди на Samsung во која се запишуваат сите апликациски активности на мобилниот уред, со датум и време. Локацијата на базата на податоци е:

  /data/data/com.android.providers.context/databases/;

- Contacts2.db – оваа база на податоци се однесува на историјата на повици и во споредба со базата кај уредите на iOS, оваа база на податоци содржи повеќе табели и многу вредни форензички информации. Локацијата на оваа база на податоци е /data/data/com.android.providers.contacts/databases/. Во оваа датотека се зачувани историјата на повик со телефонски броеви, датум на повик, времетраење на повикот и тип на повик. Доколку типот на повикот е:

  - 1 – дојдовен;

- 2 – појдовен;
- 3 – пропуштен.

Освен самата историја на повици, зачувани се податоци за колку пати е контактиран одреден телефонски број, кога последен пат е контактиран и кога била направена последната промена врз информацијата на одреден контакт содржана во телефонскиот адресар. На истата локација се наоѓа и базата на податоци profile.db, која во себе ги зачувува сите кориснички имиња кои се користат за google, facebook, за viber и др.;

- Mmssms.db – база на податоци во која се зачувани сите текстуални и мултимедијални пораки кои биле испратени, добиени или избришани. Текстуалните пораки се зачувани се во табелата Sms. Доколку колоната type има вредност 01 пораката била добиена и доколку колоната type има вредност 02, пораката била испратена. Резервна копија од табелата Sms се зачувува во mmssms.db-journal датотека. Оваа датотека може да содржи и претходно избришани пораки. Датотеката Mmssms.db ги содржи пораките кои биле избришани до пред 45 дена;
- Mailstore.<корисничко име>@gmail.com.db – оваа база на податоци содржи информации за историјата за испраќачот на и-мејлот, примачот, датум на добивање на пораката, предметот и содржината на пораката. Локацијата на базата на податоци е /data/data/com.android.providers.gmail/databases/. На оваа локација има многу датотеки кои се поврзани со веб базираната и-мејл услуга;
- Browser.db – е историја на интернет-пребарувања и во оваа база на податоци зачувани се сите посетени интернет-адреси и сите интернет-пребарувања. Локацијата на базата на податоци е: /data/data/com.android.browser/databases/. На истата локација е и датотеката webview.db во која се зачувани сите кориснички имиња и лозинки кои биле внесени при посета на одредени интернет-страници.

Уредите на Android не ги зачувуваат само текстуалните пораки, телефонскиот адресар, историјата на повици и др. Има повеќе информации отколку што може корисникот да замисли. Така, на пример, местата на кои бил со помош на GPS функционалноста.

- Geolocation.db – база на податоци во која е секогаш зачувана само последната геолокациска точка на која корисникот бил на точно запишан датум и време. Локацијата на базата на податоци е:

  /data/data/com.android.browser/gears/

- Search_history.db – содржи историја на сите пребарувања во google map апликацијата;

- CachedGeoposition.db – датотека во која се зачувани стари геолокациски податоци и е на локација: /data/data/com.android.browser/app_geolocation/

- packages.xml – xml датотека со локација на: /data/system, во која се содржани сите инсталирани апликации во мобилниот уред и информации за конфигурациските приспособувања за секоја апликација и нивните својства;

- accounts.db – база на податоци со локација на: /data/system/users/0/. Во оваа датотека се зачувани сите кориснички имиња и лозинки користени по најава на соодветна сервисна услуга како Gmail, Facebook и др.

И Andoid и iOS се способни за зачувување многу кориснички податоци и информации. Податоците се зачувани во различни датотеки на различни мемориски локации. Android е базиран на основата на Linux, но сите уреди на Android немаат ист оперативен систем и архитектура. Не е разликата само во оперативниот систем, уредите имаат и различни безбедносни механизми. Кај некои податоците се шифрирани, кај некои се заштитени со нумеричка лозинка. Често, извлекувањето на податоците при вакви активни безбедносни механизми е невозможно, особено кај уредите на iOS со хардверско шифрирање на податоците. Поради шифрирање и на нераспределениот мемориски простор,

невозможно е извлекување на избришаните податоци. Кај уредите на Android, податоците не се криптирани како кај уредите на iOS, но постои опција за активирање на таква функционалност. Исто и уредите на Android може да бидат заклучени со нумеричка лозинка која може многу лесно да биде отстранета или да се дознае која е визуелната лозинка. Сепак, не секогаш истите методи се успешни кај уред на Android од друг производител.

9. ЗАКЛУЧОК

Со брзиот развој на мобилните уреди се зголеми бројот на побарувања за форензичко вештачење на мобилните уреди. Поради ова се појави потреба за развој на процес за вештачење на уредите. Секогаш како форензичари се стремиме да бидат извлечени сите дигитално доказни податоци од мобилните уреди. Мобилниот уред iPhone на производителот Apple, значително го зголемува пазарниот удел и со 13 милиони продадени примероци станува уред за кој форензичките експерти треба да го разберат и да бидат способни за анализа. Со цел форензички да пристапиме кон iPhone, важно е да ги разбереме можностите на уредот. Многу безбедносни функции треба да се земат предвид при форензичката анализа. Концептите на анализи не се менуваат само поради претставувањето на новиот оперативен систем и хардвер. Ќе се промени само технологијата која треба да се разбере. Пазарот за форензички софтвер се бори за да имплементира поддршка за iPhone во веќе постоечките форензички програми или со создавање нова алатка која ќе обезбеди форензика на iPhone. Целта на овој магистерски труд е да прикажеме функционален форензички метод за извлекување податоци од уреди на iOS со помош на open source алатки. Целиот процес е предизвик, бидејќи од истиот производител постојат различни модели со различна верзија на оперативниот систем. Со честото објавување на новите верзии на оперативниот систем iOS, се направија драстични промени од iOS 4 и iOS 5, односно се промени и локацијата и датотеките каде дигиталните докази можат да бидат најдени. Оперативниот систем iOS е единствен во однос на другите уреди и затоа е потребно да се имаат вистинските алатки, но и претходни обуки за начинот на функционирање на уредите на iOS.

10. КОРИСТЕНА ЛИТЕРАТУРА

[1] Hoog, Andrew, and Katie Strzempka. "*iPhone and iOS Forensics: Investigation, Analysis and Mobile Security for Apple iPhone, iPad and iOS Devices"*. Elsevier, 2011;

[2] Morrissey, Sean. *iOS Forensic Analysis: for iPhone, iPad, and iPod touch*. Apress, 2010;

[3] Gladyshev, Pavel. *Formalising event reconstruction in digital investigations*. Diss. University College Dublin, 2004;

[4] Ayers, Rick, Sam Brothers, and Wayne Jansen. "Guidelines on Mobile Device Forensics (Draft)". *NIST Special Publication* 800 (2013): 101;

[5] Ahmed, El-Rabbany. "Introduction to GPS: the global positioning system". *Artech House communication series, Library of Congress, United States of America*. 2002;

[6] http://www8.garmin.com/aboutGPS/

[7] Apple, iOS Security, Oct 2012 (достапно за преземање на линкот: http://www.apple.com/ipad/business/docs/iOS_Security_Oct12.pdf);

[8] Jonathan Zdziarski, iOS Forensic Investigative Methods, 2012;

[9] Varsalone, Jesse. *Mac OS X, iPod, and iPhone forensic analysis DVD toolkit*. Syngress, 2008;

[10] Apple, I. "iOS technology overview". *iOS Developer Library. Cupertino, CA, USA: Apple Inc* (2010);

[11] Rick Ayers, Wayne Jansen, Nicolas Cilleros, Ronan Daniellou, "Cell Phone Forensic Tools: An Overview and Analysis" (2005) (достапно за преземање на линкот: http://csrc.nist.gov/publications/nistir/nistir-7250.pdf);

[12] Wayne Jansen, Rick Ayers, "Guidelines on Cell Phone Forensics" (2007);

[13] Tim Proffitt, "Forensic Analysis on iOS Devices" (2012);

[14] Mats Engman, "Forensic investigation of Apple's iPhone" (2013) (достапно на Линкот: http://www.diva-portal.org/smash/get/diva2:651693/FULLTEXT01.pdf);

[15] Bader, M., & Baggili, I. (2010). "iPhone 3GS forensics: Logical analysis using Apple iTunes backup utility";

[16] Satish B, “Forensic analysis of iPhone backups” (достапно на линкот: http://www.exploit-db.com/wp-content/themes/exploit/docs/19767.pdf);

[17] Jonathan A. Zdziarski, “iPhone/iPod Touch Forensics Manual” (2008);

[18] http://code.google.com/p/iphone-dataprotection/

[19] http://www.iclarified.com/750/where-to-download-iphone-firmware-files-from

[20] http://code.google.com/p/networkpx/downloads/detail?name=ldid&can=2&q

[21] https://github.com/downloads/osxfuse/osxfuse/

[22] http://mercurial.selenic.com/

[23] http://fr0st.zxq.net/saffron/BootRa1nPage.html

[24] http://www.yunqa.de/delphi/doku.php/products/sqlitespy/index

[25] http://www.evigator.com/free-apps/

[26] http://www.evigator.com/free-apps/

[27] http://www.bvsystems.be/photoGPSextract.php

[28] Andrew Hoog, “Android Forensics: Investigation, Analysis, and Mobile Security for Google Android” (2011).