Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Zabezpečené zpracování medicínských obrazových dat

Plzeň, 2017 Jaroslav Malát

Prohlášení

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 23. června 2017

Jaroslav Malát

Poděkování

Úvodem bych chtěl poděkovat vedoucí mé bakalářské práce doc. Dr. Ing. Janě Klečkové za důležité připomínky a rady k formální i obsahové stránce práce.

Abstract

Theme: Secure processing of medical image data

The presented bachelor's thesis is focused on the problem of security in the processing of medical image data.

The theoretical part of bachelor thesis is focused on legal regulations and laws for handling private data about patients and presents the most widely used standard for image files, which is DICOM. There is also analysis of available methods for anonymizing image data and the proposed algorithm that improves the process of anonymization.

The practical part is focused on the implementation of algorithm for anonymizing image part of medical data. The program is developed in programming language Java and I operates on system Linux Ubuntu 14.04LTS.

Abstrakt

Téma: Zabezpečené zpracování medicínských obrazových dat

Předkládaná bakalářská práce se zaměřuje na problém bezpečnosti v oblasti zpracování medicínských obrazových dat.

Teoretická část bakalářské práce se zaměřuje na právní předpisy a zákony pro práci s citlivými údaji o pacientech a seznamuje s nejpoužívanějším standardem pro obrazové soubory, kterým je DICOM. Dále je provedena analýza dostupných metod anonymizace obrazových dat a navržení algoritmu, jenž zlepšuje proces anonymizace.

Praktická část je zaměřena na implementaci navrženého algoritmu pro anonymizaci obrazové části medicínských dat. Program je vyvíjen v programovacím jazyce Java a na operačním systému LINUX UBUNTU 14.04LTS.

Obsah

Úvod

Jako téma bakalářské práce jsem si zvolil téma zabezpečené zpracování medicínských dat. Hlavním důvodem byl fakt, že medicínská data jsou součástí běžného života každého jednotlivce.

Práci jsem rozdělil do dvou částí, a to části teoretické a praktické. V teoretické části jsem se snažil o vysvětlení všech pojmů spojených se zvoleným tématem. Na úvod provedu výtah z legislativy České republiky platné ke květnu roku 2017, která se dotýká práce s citlivými daty. Dále se věnuji vymezení základních pojmů, jako je DICOM. Praktickou část jsem se rozhodl věnovat navržení a vytvoření anonymizačního programu. V poslední části této bakalářské práce jsem se věnoval zhodnocení celého mého snažení a zhodnocení funkčnosti anonymizačního programu.

Mezi cíle této práce patří analýza legislativních a bezpečnostních požadavků pro zpracování medicínských dat. Bude popsán Standardní formát souvisejících obrazových dat DICOM a také technologie PACS, umožňující správu, ukládání a zobrazování obrazové dokumentace. Dále bude provedena analýza dostupných metod anonymizace obrazových dat a navržen algoritmus zlepšující proces anonymizace. Můj anonymizační program vyhodnotí, zda obrazový soubor obsahuje jakákoliv citlivá data, ať už o lékaři nebo o pacientovi, odpovídající předem daným předpisům a následně tato data anonymizuje.

Realizace je rozdělena do následujících bodů, které jsou stejné jako body v zadání bakalářské práce a to:

- seznámit se se současnými legislativními a bezpečnostními požadavky týkajícími se zpracování medicínských dat a standardním formátem souvisejících obrazových dat,
- provést analýzu dostupných metod anonymizace obrazových dat a navrhnout algoritmus zlepšující proces anonymizace,
- navržený algoritmus implementovat a ověřit jeho správnou činnost,
- výsledky zhodnotit.

Práce by měla sloužit jako ucelený popis zabezpečeného zpracování medicínských dat.

1. Úvod do bezpečnostní problematiky a obrazová dokumentace

Na úvod bych rád definoval pojem zdravotnické dokumentace. Zdravotnická dokumentace je souhrn informací o pacientovi (klientovi) zdravotnického zařízení, který může být vedený v jakékoliv podobě.

Za termínem „citlivá data“ si představuji ta data, které poukazují na osobní údaje pacienta nebo vykonávajícího lékaře. Také se mohou považovat za citlivá data i místo, datum a čas pořízení snímku vystření, neboť se díky těmto údajům může snadno zpětně dohledat původní (originální) snímek obrazové dokumentace.

Tato dokumentace má především sloužit jako pracovní nástroj při léčbě, ale případně i jako doklad, či dokonce důkaz v případě forenzního projednávání postupu lékaře při léčení. Nesprávně vedená dokumentace může pomoci v utvrzení o chybném postupu nebo přinejmenším znemožnit dokázání postupu správného.

Většina zdravotnických zařízení ať prvního styku, mezi které patří praktický lékař pro dospělé, zubní lékař resp. stomatologická ambulance nebo praktický lékař pro děti a dorost - pediatrie, gynekologická ambulance, lékařská služba první pomoci; dále ambulantních zařízení, do kterých se řadí oční lékař resp. oftalmologická ambulance, ortopedie, psychiatrie, neurologie, dermatologie, rehabilitace, urologie, klinická psychologie a logopedie, ORL – otorhinolaryngologie, alergologie, dermatovenerologie a zdravotní rehabilitace, nebo hospitalizačních zařízení, kam se řadí nemocnice, porodnice, nemocnice následné péče, fakultní nemocnice, léčebna dlouhodobě nemocných, odborný léčebný ústav, psychiatrická léčebna, nevyjímaje lékárny, laboratoře a lázeňská zdravotní zařízení, dnes využívají informační systémy s údaji o pacientech včetně jména a adresy, rodného čísla, platebních informací a zejména pak citlivé údaje o průběhu léčby.

Tato data jsou nesporně mnohonásobně více ohrožena oproti papírové formě. Slabými místy při procesu nakládání se zdravotnickými informacemi nejsou technologie, ale lidský prvek, který bývá často opomíjen.

Z konkrétních případů lze uvést například hackerský útok na informační síť nejmenovaného zdravotnického zařízení. Pokud půjdeme do krajnosti, mohla by být ohrožena i celková zdravotnická péče. Na druhou stranu můžeme tvrdit, že absolutní bezpečnost informačních systémů je vždy pouze teoretickým pojmem.

Jako argument uvádím zjištění z kontrolní činnosti ÚOOÚ (Úřadu pro ochranu osobních údajů), více na (1).

- Elektronická podoba zdravotnické dokumentace nebyla totožná s tištěnou.
- Informační systém nemocnice neumožňoval aktivní verzi sledování přístupu do něj, což je ze zákona povinné a musí to být zaznamenáváno.
- Do informačního systému nemocnice vstoupila neoprávněná osoba, přičemž ji nebylo možno ověřit kvůli vypnuté funkci monitorování přístupu.
- Zveřejnění záznamu z operačního zákroku na webu nemocnice. Byla zřejmá identifikace osoby a k tomu byl ještě záznam doplněn jménem, částí příjmení, a dokonce rodným číslem. I přes doklad o písemném souhlasu musel být záznam okamžitě odstraněn.
- Došlo k úniku citlivých informací o pacientovi. Lékař neznající bezpečnostní prvky hesel komunikoval s pacienty přes e-mail s velmi jednoduchým a nedostatečným heslem, které bylo prolomeno.
- Dále také případ nestátního zdravotnického zařízení, které mělo registrační skříně v čekárně pro pacienty. V ordinaci se střídaly dvě lékařky a nedocházelo k důslednému zamykání skříní. Výsledkem bylo, že kdokoliv v čekárně mohl mít snadný přístup ke zdravotnické dokumentaci ostatních pacientů.

Dalším rizikovým faktorem je bezesporu to, že k citlivým datům přistupují nejen zdravotníci, ale i různé dodavatelské firmy, správci informační sítě apod.

Dále je k datům vyžadován četnější přístup ve srovnání s klasickou papírovou dokumentací.

Jak jsem již řekl, bohužel nelze docílit dokonalé ochrany informačního systému, ale snahou by mělo být dosáhnout optimální úrovně zabezpečení. Zdravotnická zařízení by měla mít vypracován plán kybernetické bezpečnosti a plán krizové připravenosti, který popisuje možné rizikové situace jak vně perimetru (mimo

nemocnici – živelná pohroma, teroristický útok, velká dopravní nehoda), tak uvnitř (požár, teroristický útok). Kybernetický útok je nebezpečí hrozící stále většímu množství lidí. Zvenku je to především možnost průniku internetem, WI-FI sítěmi nebo GSM sítí z chytrých zařízení. Zevnitř jde o lidský prvek - nevzdělaného nebo nedisciplinovaného uživatele. Ve většině případů zdravotnických zařízení nejde pouze o snahu ochránit osobní údaje pacientů, ale také o udržení jejich chodu. V tomto ohledu zůstávají zdravotnická zařízení mnohem citlivějším místem k útoku než například státní instituce či banka. Velmi důležité je školit uživatele IT systémů, a tak eliminovat případné chyby lidského faktoru.

Dalšími důvody většího zabezpečení dat ve zdravotnictví lépe než v jiných oborech lidské činnosti je například chybná diagnóza na základě pozměněných údajů, následné pochybení v léčbě a bezprostřední ohrožení zdraví i života samého.

Požadována je dostupnost jak životně důležitých dat a údajů v případě ohrožení zdraví, či přímo života pacienta, tak dostupnost údajů pacienta různými odděleními, dále dostupnost pro služby, změny personálu, zástupy a rozlišení různé citlivosti dat pacienta.

Aby ochrana byla účinná, měli bychom znát potenciální slabá místa a možné útočníky (vnitřní i vnější), míru ohrožení, případné postupy, nutné náklady na eliminaci rizik a typ hrozby či útoku, jaká IS vrstva je ohrožena (infrastruktura, OS, DB, aplikace), výstupy mimo informační systém zdravotnického zařízení a způsoby zabezpečení výměny dat mezi zdravotnickými zařízeními, pojišťovnami a dalšími subjekty.

1.1. Právní předpisy v ČR

Z hlediska práce se zdravotnickou dokumentací jsou v české legislativě v současnosti tyto důležité platné zákony a vyhlášky:

- zákon č. 96/2001 Sb., o lidských právech a biomedicíně,
- zákon č. 372/2011 Sb., o zdravotních službách,
- vyhláška č. 98/2012 Sb., o zdravotnické dokumentaci,
- zákon č. 101/2000 Sb., o ochraně osobních údajů,
- zákon č. 181/2014 Sb., o kybernetické bezpečnosti,

- vyhláška č. 316/2014 Sb., o kybernetické bezpečnosti,
- vyhláška č. 317/2014 Sb., o významných informačních systémech a jejich určujících kritériích.

Více informací o zákonech na (2) nebo o kybernetickém zákoně na (3).

Z výše uvedených zákonů a vyhlášek se pokusím vybrat či citovat pasáže, které se přímo či okrajově dotýkají tématu bakalářské práce.

1.1.1. Zákon č. 96/2001 Sb., o lidských právech v biomedicíně

Každý má právo na ochranu soukromí ve vztahu k informacím o svém zdraví.

Každý je oprávněn znát veškeré své informace shromažďované o jeho zdravotním stavu a je nutno respektovat i přání každého nebýt takto informován. (4)

1.1.2 Zákon č. 372/2011 Sb., o zdravotních službách a jeho novela

Část šestá zákona se věnuje přímo zdravotnické dokumentaci a národnímu zdravotnickému informačnímu systému.

Hlava první určuje zapracování osobních údajů, hlava druhá zdravotnickou dokumentaci, její vedení a nakládání s ní, možnosti do jejího nahlížení či pořizování kopií.

Hlava třetí je pak věnována přímo Národnímu zdravotnickému informačnímu systému (NZIS).

NZIS je určený ke zpravování údajů o zdravotním stavu, zdravotnických pracovnících, také vede Národní zdravotní registr, Národní registr poskytovatelů a zdravotnických pracovníků, pro potřeby výzkumu a vědy ve zdravotnické oblasti, zpracování výběrových šetření o potřebách zdravotních služeb atd. (5)

V souvislosti s anonymizací medicínských dat musím aktuálně zmínit hojně diskutovanou novelu o zdravotních službách, která nabyla účinnosti v červnu minulého roku (2016).

Dne 17.5.2016 byl ve Sbírce zákonů zveřejněn pod č. 147/2016 Sb. zákon, kterým se mění zákon č. 372/2011 Sb., o zdravotních službách a podmínkách jejich poskytování (zákon o zdravotních službách), ve znění pozdějších předpisů.

Zpracovatelem návrhu novelizačního zákona je Ministerstvo zdravotnictví, které jako hlavní cíle novely označilo jednoznačné definování působnosti Ústavu zdravotnických informací a statistiky ČR, jakožto správce Národního zdravotnického informačního systému s přesně vymezenými kompetencemi a povinnostmi.

Nově je stanoven obsah a funkčnost Národního registru zdravotnických pracovníků, či vymezení Národního registru hrazených zdravotních služeb. Nově jsou rovněž ustanoveny Národní diabetologický registr a Národní registr intenzivní péče.

Konkrétně se zdravotnickou dokumentací a zdravotnickým informačním systémem zaobírá část šestá zákona, hlava I. - zpracování osobních údajů.

Zde je důležitý zejména § 52.

Při zpracování osobních údajů lze nakládat s rodným číslem pacienta.

Hlava II. řeší zdravotnickou dokumentaci, její vedení a ukončení v různých případech.

Hlava III. řeší Národní zdravotnický informační systém, který je určený mimo jiné pro zpracování údajů o zdravotním stavu obyvatelstva.

Údaji k identifikaci pacienta mohou být např.: číslo pojištěnce, rodné číslo, datum narození, popř. část adresy.

§ 73 říká, že pro statistické a vědecké účely poskytuje statistický ústav z národních zdravotních registrů údaje pouze v podobě, z níž nelze určit konkrétní osobu, ať fyzickou či právnickou.

Národního registru reprodukčního zdraví jsou zpracovány osobní údaje, jenž jsou potřebné pro identifikaci těhotné ženy, rodiček, nenarozeného dítěte, ženy s umělým či samovolným přerušením těhotenství. V registru jsou zpracovány údaje o reimplantačních a prenatálních vyšetřeních, taktéž i o potratech.

V Národním diabetologickém registru jsou mimo jiné zpracovány i rizikové i prognostické faktory onemocnění, údaje k léčbě, osobní a rodinné anamnézy atd.

Osobní údaje jsou anonymizovány po 25 letech od úmrtí. V národním registru intenzivní péče již po 5 letech. (6)

1.1.3. Vyhláška č. 98/2012 Sb., o zdravotnické dokumentaci

V § 1 se určuje, co má obsahovat zdravotnická dokumentace (identifikační údaje poskytovatele a pacienta, pacientovo pohlaví, data zápisu, razítka, pracovní závěry a konečnou diagnózu, rozsah poskytnutých služeb, aktuální vývoj zdravotního stavu pacienta, návrh léčebných postupů, podání léčivých přípravků, lékařské posudky, záznam o pracovní neschopnosti…)

§ 2 definuje součásti zdravotnické dokumentace, § 3 co musí být uvedeno na každém listu zdravotní dokumentace a kdo je zodpovědný za zápis.

V dalších paragrafech pak nalezneme povinnosti k uchování zdravotní dokumentace a nutnost elektronického podpisu v elektronické ZD.

§ 6 ukládá, že technické prostředky pro vedení zdravotnické dokumentace v elektronické podobě zaručí:

- zabezpečení výpočetní techniky hardwarovými a softwarovými prostředky před přístupem neoprávněných osob ke zdravotnické dokumentaci,
- vedení evidence všech přístupů ke zdravotnické dokumentaci včetně jejich oprav, změn a mazání.

Příloha č. 1 k vyhlášce č. 98/2012 Sb. určuje minimální obsah samostatných částí zdravotnické dokumentace, přílohy 2 a 3 určují zásady pro dobu a samotné uchování a následné zničení ZD. (7)

1.1.4. Zákon č. 101/2000 Sb., o ochraně osobních údajů

Důvodem vzniku zákona o ochraně osobních údajů bylo Listinou lidských práv a svobod zaručené právo na ochranu občana před neoprávněným zasahováním do jeho soukromého a osobního života neoprávněným shromažďováním, zveřejňováním nebo jiným zneužíváním osobních údajů.

Zákon se vztahuje na osobní údaje, které zpracovávají státní orgány, samospráva, fyzické a právnické osoby automatizovaně nebo jinými prostředky. Nevztahuje se na zpracování údajů fyzickou osobou nebo pro osobní potřebu a ve vymezených případech též na zpravodajské služby a policii.

V §13 jsou tyto důležité údaje týkající se ochrany osobních údajů:

- Správce a zpracovatel jsou povinni přijmout taková opatření, aby nemohlo dojít k neoprávněnému přístupu k osobním údajům, k jejich změně, ztrátě nebo zničení, neoprávněným přenosům, k jinému neoprávněnému zpracování nebo i k jinému zneužití osobních údajů, přičemž tato povinnost platí i po ukončení zpracování osobních údajů.
- Správce nebo zpracovatel je povinen zpracovat a dokumentovat přijatá a provedená technická a organizační opatření k zajištění ochrany osobních údajů v souladu se zákonem a jinými právními předpisy.
- Zpracovatel nebo správce posuzuje rizika týkající se:
    - plnění pokynů pro zpracování osobních údajů osobami, které mají bezprostřední přístup k osobním údajům,
    - zabránění neoprávněným osobám přistupovat k osobním údajům a k prostředkům pro jejich zpracování,
    - zabránění neoprávněnému vytváření, úpravě, čtení, kopírování, přenosu nebo vymazání záznamů obsahujících osobní údaje,
    - opatření, která umožní ověřit a určit, komu byly osobní údaje předány.
- V oblasti automatizovaného zpracování osobních údajů je zpracovatel nebo správce v rámci opatření podle odstavce 1 povinen dále:
    - zajistit, aby systémy pro automatizovaná zpracování osobních údajů používaly pouze oprávněné osoby,
    - zajistit, aby fyzické osoby oprávněné k používání systémů pro automatizovaná zpracování osobních údajů měly přístup pouze k osobním údajům odpovídajícím oprávnění těchto osob, a to na základě zvláštních uživatelských oprávnění zřízených výlučně pro tyto osoby,

- pořizovat elektronické záznamy, které umožní určit a ověřit, kým, kdy a z jakého důvodu byly osobní údaje zaznamenány nebo jinak zpracovány,
- zabránit neoprávněnému přístupu k datovým nosičům. (8)

1.1.5. Zákon č. 181/2014 Sb., o kybernetické bezpečnosti

Předmětem úpravy tohoto zákona jsou práva a povinnosti osob, působnost a pravomoci orgánů veřejné moci v oblasti kybernetické bezpečnosti, nevztahuje se na informační nebo komunikační systémy, jež nakládají s utajovanými informacemi.

Zákon je platný do 30.6.2017 poté bude platná novela zákona č. 104/2017 Sb..

Hlava II popisuje systém zajištění kybernetické bezpečnosti. Ten zahrnuje bezpečnostní opatření, kybernetickou bezpečnostní událost a kybernetický bezpečnostní incident a jeho hlášení.

Dále pak popisuje evidence, opatření, varování, reaktivní a ochranná opatření a možné kontaktní údaje. Vymezuje úkoly národního i vládního CERT a jeho provozovatele. CERT (Computer Emergency Response Team) vznikl v roce 1988 na základě aféry s jedním z prvních počítačových červů, kterým byl tzv. Morrisův červ, jenž využil k svému šíření celosvětové sítě internetu. Od té doby CERT monitoruje všechny internetové průlomy, informuje o zranitelných místech v různých systémech a na základě toho zveřejňuje maximální množství bezpečnostních rad.

Hlava III se pak přímo zaobírá stavem kybernetického nebezpečí. Definuje jej jako stav, kdy je ve velkém rozsahu ohrožena bezpečnost informací v informačních systémech nebo bezpečnost a integrita služeb nebo sítí elektronických komunikací. Nalezneme zde i podmínky pro vyhlášení nouzového stavu.

Všechny prováděcí předpisy k zákonu č. 181/2014 Sb., o kybernetické bezpečnosti, které platí stejně jako zákon od 1. 1. 2015, byly dne 19. 12. 2014 uveřejněny ve Sbírce zákonů v částce 127 pod tímto označením:

317/2014 Vyhláška o významných informačních systémech a jejich určujících kritériích

316/2014 Vyhláška o bezpečnostních opatřeních, kybernetických bezpečnostních incidentech, reaktivních opatřeních a o stanovení náležitostí podání v oblasti kybernetické bezpečnosti (vyhláška o kybernetické bezpečnosti)

315/2014 Nařízení vlády, kterým se mění nařízení vlády č. 432/2010 Sb., o kritériích pro určení prvku kritické infrastruktury. (9)

1.1.6. Vyhláška č. 316/2014 Sb., o kybernetické bezpečnosti

Touto vyhláškou se stanovuje obsah a struktura bezpečnostní dokumentace pro informační systém kritické informační infrastruktury, významný informační systém, komunikační systém kritické informační, obsah bezpečnostních opatření, rozsah jejich zavedení, typy a kategorie kybernetických bezpečnostních incidentů, náležitosti a způsob hlášení kybernetického bezpečnostního incidentu, náležitosti oznámení o provedení reaktivního opatření a jeho výsledku a vzor oznámení kontaktních údajů a jeho formu.

Při hodnocení rizik se zvažují například tyto hrozby:

- porušení bezpečnostní politiky, zneužití oprávnění ze strany administrátorů a uživatelů, provedení neoprávněných činností,
- selhání nebo poškození technického anebo programového vybavení,
- zneužití identity fyzické osoby,
- nedostatky při poskytování služeb informačního systému kritické informační infrastruktury, významného informačního systému nebo komunikačního systému kritické informační infrastruktury,
- neoprávněná modifikace nebo zneužití údajů,
- poškození nebo odcizení aktiva. (10)

1.1.7. Vyhláška č. 317/2014 Sb., o významných informačních systémech a jejich určujících kritériích

Tato vyhláška nepřímo souvisí s tématem a problematikou anonymizace obrazových medicínských dat, neboť při narušení bezpečnosti informačního systému by dále nemělo smysl anonymizovat obrazová medicínská data.

Ministerstvo vnitra a Národní bezpečnostní úřad stanoví podle § 28 odst. 1 zákona č. 181/2014 Sb., o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti).

Touto vyhláškou se stanoví významné informační systémy a jejich určující kritéria, která se člení na dopadová určující kritéria a oblastní určující kritéria.

Dopadovým určujícím kritériem je skutečnost, že úplná nebo částečná nefunkčnost informačního systému způsobená narušením bezpečnosti informací by mohla mít negativní vliv mimo jiné na provoz jiného významného informačního systému využívajícího služeb hodnoceného informačního systému, který je nefunkční, a dále oběti na životech s mezní hodnotou více než 10 mrtvých nebo 100 zraněných osob vyžadujících lékařské ošetření, s případnou hospitalizací s dobou delší než 24 hodin. (11)

1.2. Právní předpisy v zahraničí

Za zmínku také stojí legislativa v zahraničí. Například ve Spojených státech Amerických mají zákon zvaný HIPAA.

1.2.1 HIPAA

HIPAA (Health Insurance Portability and Accountability Act), je zákon platný od roku 1996, který se používá jako mezinárodní standard. Tento zákon se zaměřuje na bezpečnost a přístupnost medicínských záznamů a dat ať už ve fyzické formě nebo ve formě digitální.

Lze ho rozdělit do 5 částí (hlav):

- hlava 1 – ochraňuje zdravotní pojištění osobám, které přišly o práci nebo práci změnily,
- hlava 2 – se zabývá předcházením podvodů se zdravotnickou péčí nebo zneužití informací, zadává národní normy na zpracování elektronické zdravotnické dokumentace,
- hlava 3 – zadává standard množství peněz, které mohou být ušetřeny za osobu v
- hlava 4 – definuje reformu zdravotního pojištění,

- hlava 5 – ustanovuje o životním pojištění a péče o ty, kteří přišli o americké občanství.

Dává právo pacientovi vyžádat si zdravotnickou dokumentaci nebo její kopii, navrhnout změny, zamítnou nebo povolit sdílení informací různými osobami (např. rodina).

Zákon nařizuje zdravotnickým organizacím, aby uchovával zdravotnickou dokumentaci pacientů dostatečně zabezpečenou. Za porušení bezpečnosti hrozí sankce.

Pacienti jsou při návštěvě doktora vyzváni k podepsání tzv. Notice of Privacy Practices, což je formulář informující o tom, jak konkrétní zdravotnické zařízení nakládá se zdravotnickou dokumentací pacientů.

Zdravotnická zařízení by se měla dotazovat pacienta, zda chce sdílet zdravotnickou dokumentaci s jinou organizací nebo osobou. Toto pravidlo ale neplatí ve všech případech, např. lékař může sdělit zdravotní stav druhému ošetřujícímu lékaři bez zeptání se pacienta. To platí i pro sestřičky, které se o pacienta starají. Ošetřující osoba (lékař nebo sestřička) nemá právo nahlížet do zdravotnické dokumentace pacientů, o které se nestará, hrozí za to pokuty či případná ztráta zaměstnání.

K zajištění dodržování zákonu HIPAA jsou v nemocnicích kontroly, které sledují přístup všech zaměstnanců zdravotnického zařízení, aby se zjistilo, kdo a jak nakládal se zdravotnickou dokumentací pacienta. Slouží to poté jako důkaz. (12)

Důležité je také zmínit pojem PHI (Protected Health Information) a ePHI (electronic Protected Health Infromation), což je jakákoliv informace o zdravotním stavu, poskytování zdravotní péče nebo platba za zdravotní péči, která může být spojena se specifickou osobou (pacientem).

Tyto informace podléhají anonymizaci podle zákonu HIPAA. Rozdělují se na 18 identifikátorů:

1. jména,
2. geografická data,
3. datumy všeho druhu,

4. telefonní čísla,
5. čísla faxů,
6. emailové adresy,
7. číslo sociálního zabezpečení,
8. čísla lékařských záznamů,
9. čísla příjemců zdravotního plánu,
10. čísla účtů,
11. čísla certifikátů nebo licencí,
12. identifikátory vozidel, jejich sériová čísla nebo poznávací značky,
13. identifikátory zařízení a jejich sériová čísla,
14. webové adresy,
15. adresy internetového protokolu (IP adresy),
16. biometrické identifikátory (otisky prstů, skenování sítnice),
17. fotografie obličejů nebo fotky k nim přirovnatelné a
18. jakékoliv unikátní identifikační číslo nebo kód.

Při zacházení s těmito údaji musí každý dodržovat bezpečnostní pravidla HIPAA (Security Rule). (13)

1.3. DICOM

Na úvod bych rád uvedl, že anonymizace formátu DICOM již byla v předešlých letech vyřešena, a tedy přímo s formátem DICOM nepracuji, pouze s jeho obrazovou částí, tj. vyexportovaného souboru typu PNG. Přesto si myslím, že formát DICOM s tématikou úzce souvisí a bylo by vhodné tento formát alespoň obecně popsat.

DICOM (Digital Imaging and Communications in Medicine) je mezinárodní standard pro komunikaci a správu obrazových medicínských dat a data s nimi spojená (ISO 12052). Definuje formáty pro obrazová medicínská data, aby mohla být posílána v kvalitě nezbytné pro lékařské účely.

DICOM můžeme najít v každém radiologickém, kardiologickém, a i v zařízení pro radioterapii, mezi ně patří například X-ray. Zvyšuje se však využití i v dalších oblastech lékařství, například v očním a zubním lékařství.

DICOM, jak již bylo zmíněno, je standard, který popíše, přenese a uloží informace vzniklé na modalitách. DICOM využívají všechny zobrazovací modality, mimo již uvedených – skiagrafie, mamografie, výpočetní tomografie (CT), magnetická rezonance (MR), ultrazvuk, pozitron emisní tomografie atd.

DICOM mimo jiné popisuje, jak má být komprimována (zhuštěna) určitá obrazová informace, jakých má být použito metod a to jak na straně lékařského zařízení (modality), tak na straně stanice.

S desítkami tisíc zobrazovacích zařízení v provozu a s miliardami lékařských snímků se stává DICOM jedním z nejrozšířenějších zdravotních standardů po celém světě. (14)

1.3.1. Historie vzniku DICOM

Když bylo poprvé představeno CT společně s dalšími digitálními diagnostickými zobrazovacími metodami a se stále rostoucím využíváním počítačů pro klinické aplikace, ACR (American College of Radiology) a NEMA (National Electrical Manufacturers Association), vyvstala nutně potřeba vytvoření standardu pro přenos snímků a s nimi souvisejících informací, a to i mezi zařízeními od různých výrobců.

ACR a NEMA tedy vytvořily společný výbor v roce 1983, aby vytvořily standard, který by:

- podporoval komunikaci digitálních obrazových dat bez ohledu na výrobce zařízení,
- usnadnil rozvoj a rozšíření archivace obrazu a komunikačních systémů (PACS), které mohou také komunikovat s jinými systémy nemocničních informací,
- umožnil vytvoření diagnostických informačních databází, které mohou být čteny velkým rozsahem geograficky distribuovaných zařízení.

Od prvního zveřejnění v roce 1993, způsobil DICOM revoluci v praxi radiologie, kdy bylo možné vyměnit X-ray filmy za plně digitální workflow.

Ať už u oddělení urgentního příjmu, u srdečního zátěžového testování nebo u detekce rakoviny prsu, je DICOM standard, který usnadňuje práci při komunikaci s medicínskými daty, a tím usnadňuje práci lékařům.

1.3.2. Základní části standardu DICOM

DICOM standard se původně skládal z 20 základních částí, avšak části PS3.9 a PS3.13 byly postupem času odstraněny. (15)

PS 3.2 Shoda

V této části standardu jsou definovány principy, které musí zařízení nebo informační systém splňovat, aby dosáhl shody se standardem.

- Požadavky na shodu – část PS3.2 specifikuje obecné požadavky, které musí být v procesu implementace splněny. Konkrétní požadavky pro jednotlivé funkce, data i příkazy jsou pak uvedeny ve specifických částích standardu.
- Prohlášení o shodě – část PS3.2 definuje strukturu dokumentu Prohlášení o shodě. Specifikuje informace, které musí být v dokumentu obsaženy, včetně vazeb na konkrétní požadavky uvedené v ostatních částech standardu.

PS 3.3 Definice informačních objektů

V této části standardu jsou specifikovány třídy informačních objektů (Information Object Classes), které umožňují realizovat abstraktní definici entit reálného světa aplikovatelnou při komunikaci a přenosu medicínských obrazů a informace s nimi spojené (křivky, strukturalizované nálezy, dávky radiační terapie, atd.). Každá definice třídy informačních objektů je tvořena popisem jejího určení a atributů, pomocí kterých je definice realizována.

Standard rozlišuje dva typy tříd informačních objektů:

- Normalizované třídy informačních objektů – obsahují pouze atributy, které jsou vlastní reprezentované entitě reálného světa.
- Kompozitní třídy informačních objektů – mohou obsahovat i atributy, související s entitou reálného světa, které nejsou vlastní (cizorodé).

Kompozitní třídy informačních objektů udávají strukturalizovaný rámec pro realizaci komunikačních požadavků pro zajištění úzké vazby mezi obrazovou informací a informacemi s nimi souvislými.

PS 3.4 Specifikace servisních tříd

Tato část definuje řadu servisních tříd. Servisní třída spojuje jeden nebo víc informačních objektů s jedním nebo více příkazy, které nad těmito informačními objekty mají být vykonány.

Mezi servisní třídy například patří:

- management tisku,
- uložení informací,
- dotaz/opověď,
- základní management worklistu,
- management pacienta,
- management výsledků.

PS 3.5 Datové struktury a kódování.

V této části se specifikuje vytváření a kódování datových souborů (Data set) DICOM aplikací, které vycházejí z užití informačních objektů a servisních tříd. V části se také specifikuje, jaké jsou použité kompresní techniky.

PS 3.6 Datový slovník

V této části se specifikuje centrální registr DICOM datových elementů a jejich definic. Datové elementy představují základní entitu reprezentované informace, včetně jejich unikátní identifikace v rámci standardu DICOM.

Každý datový element je specifikován:

- jednoznačným tagem, tvořeným z čísla skupiny a z čísla elementu,
- jménem,
- hodnotou multiplicity (číslo, udávající kolik hodnot může být zakódováno do datového elementu),
- typem hodnoty (integer číslo, řetězec znaků, atd.),
- každý unikátní identifikátor je specifikován - složením ze dvou částí, které jsou odděleny tečkou.
    - <org root> - unikátní číselná hodnota pro organizaci
    - <suffix> - unikátní číselná hodnota v rámci organizace

Příklad:

UID = <org root>.<suffix>

PS 3.7 Výměna zpráv

Tato část specifikuje služby a protokoly používané aplikacemi medicínských zobrazovacích metod při výměně zpráv v rámci DICOM komunikace. Tyto zprávy jsou složeny z posloupnosti příkazů a z navazujícího datového streamu.

PS 3.7 dále udává:

- operace a informace o stavu (nebo případné změně stavu) entity (DIMSE služby – DICOM Message Service Element), které jsou k dispozici jednotlivým třídám služeb definovaných v části PS 3.4,
- pravidla pro ovládání příkazů realizujících komunikaci na principu požadavek/odezva,
- pravidla pro navázání a ukončení spojení zajišťovaného komunikačními službami,
- kódovací pravidla nezbytná pro tvorbu posloupností příkazů a zpráv.

PS 3.8 Podpora síťové komunikace pro výměnu zpráv

V této části se specifikují komunikační služby a protokoly nejvyšší komunikační vrstvy nezbytné pro komunikaci mezi DICOM aplikacemi, které zajišťují, aby komunikace byla prováděna efektivně a koordinovaně v daném síťovém prostředí. Uvedená specifikace služeb vrchní komunikační vrstvy (Upper Layer Service) je podmnožinou služeb zajišťovaných sedmivrstvovým komunikačním modelem ISO/OSI. Její definice specifikuje použití protokolu DICOM horní vrstvy ve spojení s TCP/IP transportním protokolem.

PS 3.10 Paměťová média a formát souboru pro výměnu médií

Tato část specifikuje obecný model ukládání medicínských obrazových dat na výměnných médiích. Hlavním účelem této části je poskytnout rámec umožňující vzájemnou výměnu různých typů medicínských obrazových dat i s nimi souvisejícími informacemi na různé typy paměťových médií.

PS 3.11 Aplikační profily paměťových médií

V této části se specifikuje aplikační podmnožina DICOM standardu, pro kterou implementace může dosáhnout shody. Takovéto prohlášení shody je aplikováno na funkčnost procesu výměny medicínských obrazových dat a s nimi souvisejícími informacemi na paměťových médiích pro specifické klinické využití.

PS 3.12 Formáty medií a fyzická média pro výměnu medií

Tato část podporuje a usnadňuje výměnu informací mezi medicínskými aplikacemi a specifikuje:

- charakteristiku specifických fyzických medií a jejich formátů,
- strukturu pro popis vzájemných vztahů mezi obecným modelem paměťových medií a specifickými fyzickými médii a jejich formátem.

PS 3.14 Zobrazovací funkce standardní stupnice šedi

V této části se specifikují standardizované zobrazovací funkce, které jsou nezbytné pro konzistentní zobrazování obrazových dat založených na stupnici šedi. Zobrazovací funkce poskytují metody kalibrace konkrétních zobrazovacích systémů, umožňující zajistit konzistentní prezentaci obrazových dat na různých mediích (displeje, tiskárny, atd.). Zobrazovací funkce jsou založeny na lidském vizuálním vnímání (Bartenův model).

PS 3.15 Bezpečnostní a systémové profily managementu

V této části se specifikuje bezpečnost systémů DICOM standardu a pravidla řízení přístupu k datům, která musí být dodržena pro dosažení shody aplikace se standardem. Tu obstarávají obecně uznávané protokoly, jako jsou například DHCP, LDAP, TSL a další.

PS 3.16 Mapování obsahových zdrojů

Tato část DICOM standardu specifikuje, jaké návrhy formátů strukturovaných dokumentů DICOM informačních objektů lze používat. Dále také uvádí množinu kódovaných termínů, které jsou využívány informačními objekty a též překlady kódovaných termínů specifických pro jednotlivé země.

PS 3.17 Vysvětlivky

Část PS 3.14 standardu DICOM obsahuje rozsáhlé dodatečné vysvětlivky k předchozím částem. Ostatní části se na ní taktéž odkazují.

PS 3.18 Webový přístup k DICOM objektům (WADO)

V této části je specifikováno, jaké prostředky umožnují realizaci požadavku na povolené DICOM objekty ve formátu http URL/URI (Uniform Resource Locator/ Uniform Resource Identifier). Požadavek musí obsahovat směrník, který odkazuje na příslušný známý a definovaný DICOM objekt ve formě konkrétního UID.

1.4. PACS

PACS (picture archiving and communication system) je technologie ve zdravotnictví pro krátkodobou a dlouhodobou archivaci, vyhledávaní, správu, distribuci a prezentaci medicínské obrazové dokumentace. Mezi obrazovou dokumentaci řadíme snímky z rentgenu, centrálního tomografu, magnetické rezonance apod.

PACS umožňuje zdravotnické organizaci (jako je například nemocnice) zachytit, ukládat, prohlížet a sdílet všechny typy medicínské obrazové dokumentace jak interně, tak i externě. Při nasazování PACS musí zdravotnická organizace zvážit prostředí, v němž bude použito (hospitalizační, ambulantní, nouzové, specializované) a další elektronické systémy, se kterými se bude integrovat.

Interoperabilita (schopnost různých systémů vzájemně spolupracovat, poskytovat si služby, dosáhnout vzájemné součinnosti) snímků v samostatných systémech PACS je obavou pro poskytovatele zdravotní péče, a to i mezi různými poskytovateli v rámci stejného systému zdravotní péče. Přenos lékařských snímků je technologicky možný, pokud nejsou komplikované konkurenčními systémy, které nejsou interoperabilní.

Neutrální archiv dodavatele (VNA) může poskytnout jednotnou, konsolidovanou archivační platformu, pomocí níž lze hostit soubory z různých programů PACS. Větší systémy zdravotní péče se zastaralým nebo neúčinným softwarem PACS někdy zvolí implementaci podnikové VNA spíše, než modernizaci na novější PACS.

PACS má čtyři hlavní komponenty:

- imagingové systémy – magnetická rezonance (MRI), počítačová axiální tomografie (CAT scan) a rentgenové zařízení,
- zabezpečená síť – distribuce a výměna informací o pacientech,
- pracovní stanice – nebo mobilní zařízení pro prohlížení, zpracování a interpretaci obrázků,
- archivy – slouží pro ukládání a vyhledávání obrazových souborů a související dokumentace a zprávy.

Zdravotnické organizace, které instalují nebo nahrazují PACS, mohou být motivovány neefektivním pracovním postupem nebo potřebou po PACS umožňujícím zobrazovat a ukládat všechny snímky v rámci jednoho systému. PACS, podobně jako jiné systémy IT ve zdravotnictví, podléhají technologickým a regulačním změnám, které mohou donutit poskytovatele péče zvažovat novější verze. Poskytovatelé by navíc měli rozumět službám zálohování, které nabízejí jejich prodejci PACS, nebo vyvinout vlastní zásady v případech výpadků systému.

Jednou oblastní medicíny s velkým zájmem o PACS je radiologie. Radiologické PACS je často nasazeno spolu s radiologickým informačním systémem (RIS). RIS se používá k plánování schůzek pacienta a k zaznamenávání historie dat pacienta, kde se PACS zaměřuje spíše na uchovávání a vyhledávání snímků.

PACS založené na cloudovém řešení jsou typem architektury PACS, která ukládá a zálohuje medicínskou obrazovou dokumentaci zdravotnické organizace na externím serveru, nikoliv fyzicky uvnitř organizace. Uživatelé, kteří mají oprávnění k přístupu, se mohou kdykoli pomocí PACS cloudu dostat na zdravotnickou obrazovou dokumentaci. Cloud PACS také umožňuje zdravotnickému personálu zobrazit zdravotnickou obrazovou dokumentaci z jakéhokoli schváleného zařízení. (16)

Systémy Picture Archiving and Communicating System (PACS) se používají ve zdravotnictví jako podsystém nemocničního IS (informačního systému).

PACS jsou typické tím, že zpracovávají velké množství dat. Tahle objemově náročná data vznikají většinou ve specializovaných přístrojích, jako jsou centrální tomograf – CT (computed tomography), magnetická rezonance – MR (magnetic

resonance), arteriograf (XA , X-ray angiography) nebo angiograf a další. Data, jenž vznikají na těchto modalitách (digitální diagnostická zařízení používaná ve zdravotnictví pro telemedicínské sledování pacienta) mohou mít pro jednoho pacienta velikost řádově až několik gigabytů. Například při vyšetření centrálním tomografem je uděláno přes sto až tisíce obrázků o rozlišení 1024×1024 pixelů a bitové hloubce šedi cca 10-14 bitů.

Pokud jsou obrazová data v analogové formě, musíme je pro přenos do PACSu digitalizovat. Archivace fyzických snímků (hard-copy) je v tomto případě nahrazena archivací digitálních dat (soft-copy). V praxi to funguje tak, že si lékař otevře určitá data (např. CT řez, MR sken) v pracovní stanici pomocí DICOM prohlížečů, kterých je velký výběr (xVision, OsiriX, MicroDicom, ImageJ, Dicompass, Irfan View…) a které umožňují různě pracovat s daty – nastavení spektra barev, kontrastu, jasu, měření délek zkoumaného objektu atd. DICOM prohlížeče se dělí na placené a neplacené. Také se liší operačním systémem, pod kterým fungují.

Můžeme tudíž říci, že PACS je v podstatě systém, který uchová obrazové informace vzniklé na digitálních diagnostických zařízeních, která pracují na různých principech zobrazovacích metod. Zařízení jsou schopna mezi sebou komunikovat a pracovat na základě standardu DICOM.

2. Analýza dostupných metod anonymizace a návrh algoritmu

Jako dostupnou metodu anonymizace považuji program nebo softwarové řešení (dále jen anonymizery), které umožnují co nejlépe v krátké časové době a co nejspolehlivěji, tedy s co největší přesností, anonymizovat citlivá data z obrazových souborů formátu DICOM.

Existuje velké množství anonymizerů zaměřujících se na problematiku anonymizace zdravotnické dokumentace, tedy souborů formátu DICOM. Je možné je tedy rozdělit na několik skupin a podskupin. Například pod jakými operačními systémy fungují, co vše nabízejí kromě anonymizace, podle možnosti přizpůsobení anonymizace, tj. změna výstupního textu (např. „Anonymized“) a podle jednoho z nejdůležitější prvků – kolik stojí. Dále lze anonymizery dělit na programy určené pouze k anonymizaci a programy u kterých je jedna z funkcí právě anonymizace.

Po průzkumu dostupných anonymizerů jsem došel k následujícím závěrům.

Obecný úvod anonymizerů

Může se jednat relativně jednoduché programy, v některých případech dokonce pouze skripty, které slouží jen a pouze k anonymizaci citlivých dat v DICOM souborech nebo také rozsáhlé programy umožňující správu, prohlížení i úpravu souborů DICOM. Při anonymizaci hledají data určená k anonymizaci pomocí tagu.

Hojně využívaný obecný název pro anonymizer se používá „DICOM Anonymizer“ nebo různé podoby tohoto názvu. Pod tímto názvem jsem našel hned několik programů na anonymizaci, např:

a) dicomanonymizer v2.0.7
b) DICOM Anonymizer v1.5.0
c) DICOM Anonymizer v1.1.6.1
d) DICOM Anonymizer v1.8.0

Ač se na první pohled zná, že se jedná o několik verzí jednoho a toho samého programu, rozdíly jsou však mezi těmito třemi aplikacemi nečekaně velké. Obecně je teď popíši.

a) dicomanonymizer v2.0.7

Jedná se o propracovaný a přehledný program od firmy NEOLOGICA, která se specializuje na programy pracující s formátem DICOM. Tato firma má zákazníky po celém světě (Evropa, Asie, dokonce i Afrika).

Program umožňuje anonymizace jednotlivých souborů nebo celých složek. Poskytuje možnost nahradit citlivá data za libovolně zvolený text, např. „Anonym^Patient“.

Mezi podporované operační systémy patří:

- Windows 32-bit/64bit
- Mac OS X 10.6 a nižší, OS X 10.7 a vyšší
- Linux 32-bit/64-bit

Program lze stáhnout zadarmo, bohužel ale pouze trial na 5 dní, poté je potřeba zaplatit licenci, která se váže na 1 PC a to v hodnotě 49€ + daň. (17)

b) DICOM Anonymizer v1.5.0

Relativně jednoduchý program vytvořený od Wouter Veldhuis. Program je distribuovaný pod licencí GPLv3.

Program pracuje na principu přetahování („Drag’n Drop“) jednotlivých souborů či složek do spuštěné instance programu.

Mezi podporované systémy patří pouze Mac OS X 10.7 a novější. (18)

c) DICOM Anonymizer v1.1.6.1

V tomto případě se jedná o další relativně jednoduchý anonymizer. Je tvořen jedním dialogovým oknem, ve kterém lze vybrat složku obsahující DICOM soubory, tag na změnu, předponu a číselnou hodnotu, která určuje začátek rozdávání čísel.

Výsledek tedy vypadá nějak takto:

Prefix – „Case“

Začátek od čísla – „1“

Výsledek – „Case001“

Program také poskytuje možnost odstranění citlivých údajů týkajících se konkrétní zdravotnické instituce (název zařízení, stanice, jméno vykonávající doktora,…).

Mezi podporované operační systémy patří pouze MS Windows 32bit a je distribuovaný pod licencí BSD. (19)

e) DICOM Anonymizer v1.8.0

Tento anonymizer má jednoduchou a přehlednou obsluhu. Jako u ostatních lze vybírat tagy na anonymizaci a také náležitý text pro anonymizaci. Jedná se o jeden z mnoha programů vytvořený od skupiny DICOM Apps. Mezi další programy od této skupiny patří např. converter DICOM, převod DICOM na GIF nebo DICOM na JPEG a obráceně.

Podporovaný operační systém je pouze Windows.

Program lze stáhnout zadarmo, ale po 5 dnech uběhne trialová doba a program si vyžaduje zakoupení licence. Licence pro jednoho uživatele stojí 99$ a licence pro celou nemocnici stojí 999.99$. (20)

Další programy určené pro anonymizaci se už liší i v názvu. Uvedu opět jen obecně nějaké anonymizery.

- Anonymize IJ DICOM
- Conquest DICOM
- DICOM Rewriter
- LONI Inspector
- MIRC DicomEditor

Anonymize IJ DICOM

Jedná se o jednoduchý plugin, který vytvořil Julian Cooper. Slouží k anonymizaci metadat po tom, co byl soubor formátu DICOM otevřený v programu ImageJ.

Jedná se

Podporované operační systémy jsou tyto:

- Windows 32-bit/64-bit

- Linux
- Mac OS X 10.8 a novější

Plugin je zdarma a není nijak finančně zpoplatněn. (21)

Conquest DICOM

Další program je relativně složitý a rozsáhlý program ConQuest 1.4.19, který byl napsán Marcelem van Herkem a Lambertem Zijpem z Nizozemského onkologického institutu.

Program zvládá trénovat a testovat DICOM, convert obrazových souborů ze scanneru do DICOM a mnoho dalšího. Také obsahuje DICOM server.

Program lze spustit na operačních systémech:

- Windows
- Linux

(22)

DICOM Rewriter

Je obdobným řešením jako je Anonymizate IJ. Jedná se tedy o plugin, který se spouští ve složce s ImageJ. Byl vytvořený Walter O'Dell PhD. Jelikož se jedná pouze o plugin, je Rewriter značně omezený na možnostech úprav. Například je-li změna stringové hodnoty, která bude zapsána místo citlivých dat, větší než hodnota původního headeru (hlavičky), nebude zvolená stringová hodnota po zapsání zobrazena.

(23)

LONI Inspector

Program napsaný v Java od společnosti LONI (Laboratory of Neuro Imaging),

Umožnuje prohlížet metadata několika druhů formátů, mezi než patří i formát DICOM. Je i schopný hledat rozdíly v metadatach u 2 nebo víc souborů.

Umí také exportovat metadata do XML nebo CSV souborů.

Podporované systémy by měli být:

- Windows
- Mac OS X
- Linux

Spadá pod licenci LONI Software License.

(24)

MIRC DicomEditor

Poslední program je DicomEditor. Program měl být pouzíván jako testovací program pro MIRC (Medical Imaging Resource Center), ale obecně je používaný jako program na úpravu DICOM obrazových souborů.

(25)

S rostoucím vlivem sociálních sítí vznikla i poptávka po anonymizaci obrazových dat, z toho se nejčastěji týkala anonymizace fotek. V dnešní době existuje hned několik metod pro anonymizaci. Mezi časté patří tzv. browserové anonymizery, které umožňují anonymizaci kdekoliv, kde je zajištěn přístup k internetu (tedy bez nutnosti instalace anonymizeru). Jelikož pracuji pouze s obrazovou částí souboru DICOM, nepotřebuji anonymizer formátu DICOM, mohl bych tedy považovat browserové nebo lokální anonymizery za dostačující. Tyto anonymyzery ale nesplňují jedny ze základních podmínek – např. u browserových anonymizerů se může jako problém považovat zabezpečení, neboť se odesílají obrazové soubory na web hostovaný tvůrci aplikace a také se vyskytne problém při zpracování velkého množství dat. Lokální aplikace na anonymizace obrazových souborů zase nejsou automatické a je nutný zásah uživatele u každého obrazového souboru. Proto si myslím, že browserové nebo lokální anonymizery obrazových souborů nejsou vhodným a dostačujícím řešením pro anonymizaci medicínských obrazových dat. Přesto vše ale jeden uvedu.

ZORRO Anonymizace

Jedná se o browserovou metodu společnosti Atbon a.s. na anonymizaci dokumentů a začerňování objektů.

Po nahrání souborů do webové aplikace lze přes filtr vyhledat předdefinované vzory (např. rodné číslo). Aplikace nabízí anonymizovat všechny části naráz nebo postupně. Po stisknutí tlačítka Redigovat se zobrazí dokument se šablonou pro anonymizaci obrazového souboru, který byl automaticky vytvořen. Dopředu vyhledané texty pro anonymizaci lze jednoduše zrušit a požadované texty, které nebyly nalezeny automaticky v aplikaci, je poté možné přidat pomocí vykreslení obdélníku.

Pro získání finální verze anonymizovaného obrazového souboru je potřeba stisknout tlačítko Vytvořit redigovaný dokument, které nabídne uložení do počítače nebo zobrazení obrazového souboru v prohlížeči.

Podle referencí je tato metoda hojně využívaná krajskými i městskými úřady České republiky. Více informací zde (26).

Vyhodnocení

Ačkoli jsou browserové anonymizátory užitečné, nejsou vhodným řešením pro anonymizaci citlivých dat obsažených v obrazové části DICOM souborů. Anonymizery DICOM souborů zase nepodporují čtení pouze z obrazové části a zaměřují se převážně na metadata, které já ve své práci k dispozici nemám. Obrazové soubory by měly být anonymizovány diskrétně, proto považuji odesílání obrazového souboru na neznámý web za bezpečnostně neakceptovatelné řešení. Z tohoto důvodu jsem se rozhodl vytvořit vlastní anonymizační program, který bude splňovat požadavky ideálního anonymizeru, tj. bude spouštěn lokálně a bude bezplatný.

Návrh algoritmu

Ideální anonymizační program by měl, dle mého názoru, splňovat následující podmínky a vlastnosti:

Být lokální, tj. data by neměla opustit počítač, na kterém je prováděna anonymizace.

Být automatický, tj. anonymizer by neměl vyžadovat příliš mnoho operací přímo od uživatele (ideálně žádné úkony od spuštění), což je velice výhodné, zpracovává- li se velké množství obrazových dat.

Být rychlý, tj. doba strávená anonymizací jednoho obrazového souboru by měla být co nejmenší, což je opět výhodné, zpracovává-li se velké množství obrazových dat.

Být upravitelný, tj. anonymizer by měl nabízet možnosti modifikace, ať už se jedná o možnost přidání a také úpravy usnadňujících funkcí, například grafického rozhraní. Také by měl skýtat např. možnost úpravy formátu zápisu do logovacího souboru. Ve free anonymizeru nebo v placeném takové možnosti nebývají.

Být výhodný, mám teď na mysli z finančního pohledu, tedy čím levněji, tím lépe. Nejlepší by byl samozřejmě zadarmo, nikoliv však na úkor kvality. Kvalita by měla být srovnatelná s placenými anonymizery.

Být přesný, to znamená, že rozeznávání znaků by mělo být co nepřesnější. Přesnosti lze docílit úpravou rozeznávání řetězců (Tesseract-Ocr) i v anynomizaci. Mám na mysli případné trénování Tesseract-Ocr na obrazových souborech určených pro výzkum nebo na vylepšení kódu programu, pro spolehlivější ošetření citlivých dat určených k anonymizaci.

Po pečlivé analýze bodů uvedených výše jsem došel k závěru, že ideálním řešením bude vytvoření vlastního anonymizačního programu, který by měl dodržet všechny podmínky ideálního anonymizeru.

3. Analýza anonymizace obrazových dat

Modernizace způsobu nakládání se zdravotnickými obrazovými daty a neustálé vylepšování dostupné technologie pro jejich pořizování zapříčinily nutnost dalšího zpracování medicínských dat. V této práci je řešen důležitý úkol, a to odstranění zbytkových citlivých dat z obrazových souborů. Po odstranění citlivých dat jsou obrazové části DICOM souborů použitelné i mimo zdravotnická zařízení, např. pro účely výuky nebo statistiky.

3.1. Vyhodnocování citlivých údajů

Celkem jsem obdržel 86 obrazových souborů formátu PNG. V těchto 86 obrazových souborech jsou pravidla, podle kterých bude anonymizace citlivých dat probíhat.

Z uvedených 86 obrazových souborů pouze 64 obsahovalo nějaký text. Za text považuji i jediný znak.

Rozdělil jsem rozeznávání citlivých dat na dvě části:

- rozeznávání slov,
- rozeznávání čísel.

3.1.1 Rozeznávání slov

Základním pravidlem pro rozeznávání slovních citlivých údajů by mělo být, že je slovo složeno pouze z písmen abecedy. To však nelze plně uplatnit v programu, neboť se v obrazové dokumentaci vyskytují takové řetězce, které tohle pravidlo porušují. Je nutné ošetřit tyto případy rozšířením pravidla.

Příklad:

„PRIJMENI, JMENO“

Na příkladu výše lze vidět, že čárka následuje hned po příjmení pacienta nebo lékaře. Tesseract OCR tedy rozpozná, že se jedná o jedno slovo, jeden řetězec.

Dostanu tedy dva řetězce:

1. Řetězec: „PRIJMENI,“
2. Řetězec: „JMENO“

Pokud by platilo pravidlo, že slovní citlivý údaj se skládá pouze z písmen abecedy, anonymizovalo by se pouze jméno, tedy druhý řetězec, neboť příjmení je načteno správně jako jeden řetězec i s čárkou.

Bylo by možné akceptovat pouze řetězce, které končí tečkou, nebo čárkou, ale vznikl by posléze další problém v anonymizaci, viz příklad níže.

Podobná situace může nastat, je-li zapsán titul vyšetřujícího doktora, např.

„MUDR. PRIJMENI“

Tyto řetězce mohou mít 4 varianty.

První varianta těchto řetězců je, že se za titulem je tečka a mezera před jménem nebo příjmením. Příklad uveden o pár řádků výše. Rozpoznáno jako dva řetězce.

Druhá varianta těchto řetězců je, že titul je zakončen čárkou a obsahuje mezeru před jménem nebo příjmením. Rozpoznáno jako dva řetězce.

Třetí varianta je, že titul je zakončen tečkou a po ní není žádná mezera a navazuje rovnou jméno nebo příjmení. Rozpoznáno jako jeden řetězec.

Čtvrtá a poslední varianta je, že titul je zakončen čárkou a po ní není žádná mezera a navazuje rovnou jméno nebo příjmení. Rozpoznáno jako jeden řetězec.

Jako slova tedy považuji takové řetězce, které obsahují pouze písmena nebo speciální znaky, např. tečka nebo čárka.

Tečka nebo čárka se tedy může objevit kdekoliv v řetězci. Ve vzácných případech se v dokumentaci může objevit i znak stříšky (cirkumflex) „ˆ“. Jedná se o obrazové soubory z rentgenu.

Dále lze z dodaných obrazových souborů zdravotnické dokumentace nalézt náznak pravidla a tj., že ve většině případů je jméno a příjmení pacienta nebo vyšetřujícího lékaře, zapsáno pouze velkými písmenky, tedy uppercase. Tohle pravidlo však nelze použít, neboť by zhoršilo procentuální úspěšnost celkové anonymizace.

3.1.2 Rozeznávání čísel

Mezi číselné citlivé údaje lze považovat datum narození pacienta, rodné číslo pacienta, ale také i datum pořízení obrazové dokumentace nebo číslo vyšetření. Tyto poslední dvě informace totiž mohou vést k odhalení identity pacienta, je-li dostupná originální obrazová dokumentace.

Naopak hodnoty a výsledky získané z vyšetření je nutné při anonymizaci zachovat.

Při anonymizace je opět nezbytné vytvořit pravidla, která by zachovala oba dva výše uvedené požadavky.

Prvním a naprosto jasným pravidlem tedy je, že číselné citlivé údaje budou obsahovat jenom číslice. Toto pravidlo ovšem nemůže být absolutní, jelikož na výstupu z programu Tesseract-OCR se může objevit řetězec typu:

„901030/5060“

Na první pohled je jasné, že se jedná o mnou smyšlené rodné číslo pro příklad znázornění.

Výše uvedené rodné číslo obsahuje tedy sérii 11 znaků, z nichž jeden znak je znak lomítka. Pokud by platilo absolutní pravidlo, že číselné citlivé údaje mohou obsahovat v řetězci pouze číslice, nepovažoval by se tento řetězec za citlivý údaj k anonymizaci a nebyl by ošetřen ve finálním anonymizovaném obrazovém souboru. Je tedy nutno rozšířit první pravidlo a akceptovat speciální znaky.

Mezi akceptované (speciální) znaky jsem se rozhodl zařadit následující znaky:

- znak lomítka - „ / “,
- znak otazníku - „ **?** “,
- písmeno „**I**“.

Důvod, proč jsem se rozhodl pro lomítko je očividné z předchozího příkladu.

Znaky otazníku a písmena „**I**“ jsou pro ošetření chyby ze špatného načtení znaku programem Tesseract-OCR, kdy se může stát, že číslovka „**7**“ se zamění za znak otazníku „**?**“ a číslovka „**1“** může být zaměněna za písmenko „**I**“.

Dalším společným faktorem pro číselné citlivé údaje je podobná délka. Druhé pravidlo tedy je, že řetězec musí obsahovat více jak 7 znaků a zároveň nesmí mít více jak 14 znaků. Horní hranice je zavedena jako pojistka pro případ načtení nesmyslného řetězce.

V další části mojí bakalářské práce uvedu postupně metody a knihovny, které jsem při anonymizaci použil.

3.2. Metody hledání textových řetězců

Abych mohl citlivá data vymazat, potřebuji je nejprve v souboru najít. Jelikož však předem nevím přesné řetězce, hledám pouze řetězce splňující určitá kritéria.

Algoritmus hrubé síly

Algoritmus postupuje tak, že pro každou pozici v textu kontroluje, jestli v ní nezačíná hledaný řetězec. Složitost hledání v běžném textu je O(m+n).

Jelikož program přesně neví, jaké řetězce hledá, bylo nutné vytvořit pravidla pro hledání a anonymizaci řetězců. Algoritmus hrubé síly byl použit při kontrole řetězců, které anonymizovat nepotřebují, avšak splňují podmínky pro anonymizaci. Příkladem je, jsou-li všechny znaky v řetězci písmena – může se jednat o jméno, ale jedná se o informační údaj v obrazovém souboru (Ward, Operator, atd.). Vytvořil jsem tudíž pomocí frekvenční analýzy sérii řetězců, které při shodě s nalezeným textem (který splnil podmínky pro anonymizaci) nejsou považovány za citlivé informace nutné k anonymizaci a tedy jsou ponechány v anonymizované verzi obrazového souboru.

4. Implementace

Práci na anonymizaci jsem rozdělil do dvou částí:

- Úprava obrazových souborů a získání dat pro anonymizaci obrazových souborů.
- Anonymizace obrazových souborů.

Program sám obstarává obě dvě části.

Vývoj programu jsem implementoval v operačním systému Linux.

Program spouštím pomocí příkazu:

java –jar program.jar properties.properties

První parametr označuje properties soubor, ve kterém jsou uloženy všechny potřebné proměnné. Mezi potřebné proměnné patří cesta, název původního obrazového souboru pro anonymizaci, soubor obsahující tzv. „bílá slova“. „Bílá slova“ jsou slova, která mohou splňovat parametry pro anonymizaci, avšak nebudou anonymizována - např. názvy přístrojů (Phillips, SIEMENS, atd.). Dále také cesta k výstupnímu adresáři, kam bude uložen logovací soubor a anonymizovaný obrazový soubor. Do výstupní složky budou taktéž uloženy soubory potřebné pro anonymizaci, ty však budou před ukončením programu smazány.

Soubor properties.properties

Soubor properties.properties obsahuje důležité proměnné pro spuštění a správnou funkci programu. Hlavní parametry:

- nazev – umístění složky s PNG soubory k anonymizaci,
- cesta – umístění složky pro výstupní anonymizované soubory,
- test – obsahuje cestu k souboru s tzv. bílými slovy,
- cetnost – obsahuje název souboru typu JSON.

Dále rozšiřuje možnosti při úpravě obrazového souboru pro přesnější načítání znaků. Parametry lze psát pod sebe a doplnit jich hned několik. Zápis musí dodržovat příslušná pravidla.

První pravidlo je, že název parametru musí vždy začínat s „convert.“ a pokračuje s číslovkou. To samé platí pro parametry Tesseract-OCR.

Příklad:

„convert.1 = resize 5000“
„convert.2 = colorspace Gray“
„tesseract.1 = l eng“

Program si poté sám připraví proměnné při spuštění procesu ImageMagick.

4.1. Výběr programovacího jazyka

Pro zpracování požadavků bakalářské práce jsem se rozhodl použít programovací jazyk Java a to nejenom proto, že je jedním z nejrozšířenějších programovacích jazyků, ale také proto, že mám s Javou nejvíce zkušeností získaných při studiu.

4.2. Použité nástroje

Abych docílil splnění zadání své bakalářské práce, potřeboval jsem využít různé škály open source nástrojů. V následujících kapitolách vysvětlím, jaké open source nástroje jsem použil a také je stručně popíši.

Na úpravu obrazových souborů jsem použil **ImageMagick**, distribuovaný pro nejrůznější operační systémy. Po úpravě obrazového souboru jsem použil open source engine **Tesseract-ocr** pro naskenování dat z obrazového souboru a jejich následné uložení do příslušného formátu.

Jakožto vhodný formát pro výstupní data z **Tesseract-ocr** jsem zvolil formát **HOCR**, který je odnoží hypertextových formátů (**HTML**), konkrétně **XHTML**. Důvodem zvolení **HOCR** formátu bylo, že tento výstupní formát uchovává i pozice nalezeného textu, které jsou nezbytné pro jeho anonymizaci. Textový výstup nebo výstup **PDF** informace o poloze textu neuvádí.

Pro parsování formátu **HOCR** jsem ve svém programu využil Java knihovnu zvanou **JSoup**, díky které jsem byl schopen načtená data uložit do pole stringů, se kterým se mi bude lépe pracovat.

Jako vývojové prostředí pro svůj program jsem použil **NetBeans** pro Linux.

4.2.1. Tesseract-Ocr 3.03

Tesseract je pravděpodobně jedním z nejpřesnějších open source OCR. V kombinaci s Leptonica Image Processing Library je Tesseract schopen přečíst rozsáhlé množství obrazových formátů a dokáže je i převést do více jak 60 různých jazyků. Tesseract byl považován za jeden ze tří nejlepších engine v UNLV testu přesnosti v roce 1995. Od roku 1995 se vývoj Tesseractu zpomalil až do roku 2006, kdy začal být vývoj Tesseractu sponzorovaný společností Google. Nyní je vydán pod Apache Licencí 2.0.

Tesseract podporuje operační systémy:

- Windows
- Linux
- Mac OS X.

Důvodů, proč jsem si vybral Tesseract pro skenování obrazového souboru a upřednostnil ho tak před ostatními Open Source OCR enginy, je hned několik:

- Tesseract je celosvětově považován za jeden z nejlepších OCR enginů.
- Výsledky rozpoznávání znaků byly lepší než u ostatních mnou zkoušených enginů (např. GOCR).
- Tesseract lze trénovat pro lepší výsledky rozpoznávání znaků.
- Možnost výstupu dat ve formátu XHTML.

Výstupním formátem Tesseract může být textový soubor, PDF soubor nebo soubor HOCR. Pro svou bakalářskou práci jsem použil výstupní formát typu HOCR, který mi umožnil získat pozice mnou hledaných slov (stringů).

Přesnost rozlišování znaků se v mém případě pohybovala okolo 80%. Více viz (27).

4.2.2. ImageMagick 6.7.7-10

ImageMagick je balík nástrojů na vytváření, úpravu a zpracování bitmapových obrázků. Dokáže číst a přepisovat rozsáhlou škálu formátů souborů (oficiální stránky udávají přes 200 typů souborů), mezi nimiž jsou i formáty použité při zpracování bakalářské práce, tedy původní formát PNG a mnou převedený formát JPG.

Důvodem použití tohoto nástroje byla nutná úprava obrazových souborů pro zlepšení rozpoznávání znaků programu Tesseract-OCR. Viz více na (28).

4.2.3. JSoup

Při programování praktické části své práce jsem použil Java knihovnu JSoup, která slouží k parsování dat z hypertextových typů souborů, v mém případě tedy XHTML – hocr. Více informací je dostupných na (29).

4.2.4. AWT Graphics

K načtení obrazového souboru do programu a k následné anonymizaci citlivých dat jsem se rozhodl použít základní knihovnu AWT Graphics, která je standardně obsažena v Javě.

4.2.5. jTessBoxEditor 1.6

Tesseract-OCR nabízí možnost trénování pro lepší výsledky při rozeznávání textu. Pro trénování jsem použil grafické rozhraní jTessBoxEditor, jež umožňuje export dat zlepšujících proces rozeznávání.

Grafické rozhraní jTessBoxEditor lze spustit téměř na všech operačních systémech. Podporuje formáty TIFF, JPG, PDF, BMP a pro práci důležitý PNG.

Trénování textu v grafickém rozhraní je relativně jednoduché.

Více na (30).

4.3. Třídy programu

Program jsem rozdělil do 5 tříd:

- Hlavni
- Obrazek
- Anonymizer
- Retezec
- KolekceVyskytu

4.3.1. Třída Hlavni

Tato třída slouží jako hlavní a spouštěcí třída.

V hlavní metodě je vytvořen objekt anonymizer a načten soubor properties.properties, který obsahuje potřebné údaje pro anonymizaci obrazového souboru (tj. název a cestu obrazového souboru, cestu do adresáře určeného pro zápis, cestu k souboru obsahující tzv. bílá slova, atd.).

Dále jsou načteny všechny názvy obrazových souborů obsahující koncovku PNG a ty jsou uloženy. Cyklem for se prochází uložené názvy obrazových souborů a pro každý z nich se vytvoří objekt anonymizer a po jeho vytvoření je pak zavolána metoda anonymizuj(), která anonymizuje obrazový soubor. Program končí, není-li už žádný další soubor k anonymizaci.

4.3.2. Třída Anonymizer

Třída Anonymizer obsahuje objekt Anonymizer, který je složen z následujících proměnných:

- vstup – název obrázku (může obsahovat i cestu k obrázku),
- vystup – cesta pro výstupní soubory,
- vstupBilychSlov – odkazuje na cestu k seznamu bílých slov,
- vstupCetnost – odkazuje na cestu k souboru s četnosti řetězců,
- convertParams – obsahuje parametry pro spuštění ImageMagick,
- tesseractParams – obsahuje parametry pro spuštění Tesseract-OCR.

Třída obsahuje několik metod, které postupně vysvětlím:

- anonymizuj(),
- nactiJson(),
- zapisJson(),
- provnejJmeno(int i),
- zkontrolujString(int i),
- jeCisloZnak(int i, int k),
- zkontrolujCislo(int i),
- nutnaUprava(int i),
- anonymizujJmeno(int i),
- anonymizujCislo(int i),
- anonymizujDatum(int i),
- prepocti(),
- zapis(String str, int i).

Proměnná „int i“ je index do kolekce právě testovaného (anonymizovaného) řetězce.

Metoda anonymizuj()

Tato metoda je nejdůležitější metodou v třídě Anonymizer. V metodě se vytvoří objekt gson, kolekceVyskytu a obrazek s příslušnými parametry. Dále se přiřadí soubor hOCR z metody Ocr z třídy Obrazek. Soubor hOCR se parsuje pomocí knihoven Jsoup na formát UTF- 8. Je vytvořen Buffered Writter na zapisování do logovacího souboru a také scanner pro načítání tzv. bílých slov z textového souboru.

Metoda postupně načítá každé slovo ze souboru hOCR s jeho souřadnicemi. Nesplňuje-li žádné slovo pravidla pro anonymizaci, výsledek se zapíše do logovacího souboru.

Řadí se zde i kolekce kolekceVyskytu, která slouží jako pomocná jednotka při analýze anynomizovaného textu. Je seřazená sestupně, tedy o nejčetnějších slov dolů.

Metoda nactiJson()

Tato metoda zjišťuje, zdali již existuje soubor JSON obsahující kolekci výskytů. Pokud není nalezen, vytvoří metoda nový JSON do kterého bude kolekci ukládat. Pokud nalezen je, načte ho a pokračuje v přepisu původního JSON souboru.

Metoda zapisJson()

Metoda, jak již název napovídá, slouží k zapsání kolekce do JSON souboru.

Metoda porovnejJmeno(int i)

Pomocná metoda sloužící k porovnání názvu (jména) ze seznamu bílých slov. Kontrola spočívá v porovnávání jednotlivých znaků obou řetězců. Jedná se o metodu typu boolean. True se vrací, jsou-li znaky shodné v obou řetězcích a false se vrací, jedná-li se o 2 různé řetězce.

Metoda zkontrolujString(int i)

Metodu lze rozdělit na dvě hlavní části. První část kontroluje, zda se jedná o textový řetězec či nikoliv. Akceptuje taky speciální znaky, jako je „ **.** “ (tečka), „ **,** “ (čárka) nebo „ ^ “ (stříška). Důvod je již vysvětlen v analýze obrazových dat. Ve zkratce, tyto znaky se vyskytují přímo v textu a OCR je načte jako jeden řetězec.

Metoda jeCisloZnak(int i, int k)

Další pomocná metoda je typu boolean, která vrací true, pokud je znak char číslo nebo jeden ze znaků uvedených v podmínce (otazník, lomítko,…). Hodnotu False vrací, pokud je znak písmenkem abecedy (kromě r) nebo jiným speciálním znakem.

Vstupní hodnotou jsou dva integer proměnné. První proměnná je index do kolekce právě testovaného (anonymizovaného) řetězce. Druhá proměnná je pozice znaku v řetězci.

Metoda zkontrolujCislo(int i)

Metoda kontroluje, zda je vstupní řetězec složen pouze z číslic nebo ze speciálních znaků. Pokud tomu tak není, metoda se pomocí break ukončí.

V metodě také používám parser na čas a datum v různých formátech.

Jedná-li se o číslo (tedy datum narození nebo datum vyšetření), zavolá se metoda anonymizuCislo(int i).

Metoda nutnaUprava(int i)

Tato metoda slouží čistě pouze k přehledu, zda je obrazový soubor nutné anonymizovat.

Metoda volá metodu zapis(int i) a nastaví boolean hodnotu na true, tedy je nutná úprava a bude se anonymizovat.

Metoda anonymizujJmeno(int i)

Před samotnou anonymizací řetězce je tu ještě jedna pojistka, která kontroluje, zda je počet znaků roven nebo větší než 3.

Poté tato metoda nejdříve zavolá metodu zapis(int i), dále zjistí potřebné údaje k anonymizaci, jako jsou výška a šířka obrazového souboru určeného k anonymizaci. Po získání potřebných údajů se zavolá metoda pro vykreslení obdélníku s odpovídajícími souřadnicemi.

Metoda anonymizujCislo(int)

V této metodě je zavolána zapisující metoda zapis(int i) a následně jsou vypočteny souřadnice pro vykreslení obdélníku do obrazového souboru.

Metoda AnonymizujDatum(int i)

Tato metoda zavolá metodu zapis(int i) pro zapsání souřadnic do logovacího souboru a poté se „vykreslí“ pomocí metody vykresli(int X1, int Y1, int X2, int Y2) čtverec do obrazového souboru.

Metoda prepocti()

Tato metoda slouží pro přepočtení souřadnic z upraveného obrazového souboru na souřadnice obrazového souboru určeného k anonymizaci. K přepočtení souřadnic je použita matematická trojčlenka.

Metoda zapis(String str, int i)

V této metodě se zapíše nalezené slovo do logovacího souboru. Souřadnice nalezeného slova určeného k anonymizaci jsou vytaženy z pole. Metoda po získání souřadnic slova, určeného k anonymizaci, zavolá metodu Prepocti(). Po přepočtení původních souřadnic jsou nové souřadnice zapsány do logovacího souboru.

V této metodě jsem se rozhodl využít funkce switch. String proměnná „str“ je jedna ze dvou vstupních proměnných pro tuto metodu. Tato proměnná může nabýt 3 základní stavy a od nich se odvodí a přepínaní case ve switch:

- case „neni“ – znamená, že soubor neobsahuje žádná citlivá data a tedy není nutná anonymizace, vypíše na obrazovku, že nejsou nalezena citlivá data a do souboru napíše „Clean“,
- case „nutna“ – nastane v případě, nalezne-li se v obrazovém souboru text splňující podmínky pro anonymizaci, vypíše na obrazovku, že byla nalezena cilivá data a do souboru napíše „Private data“,

default – jedná se o defaultní (základní) nastavení, které znamená, že jakýkoliv jiný vstup než „neni“ nebo „nutna“ bude považován za default.

4.3.3. Třída Obrazek

Třída Obrazek obsahuje objekt Obrazek, který se skládá z proměnných:

- vstup – název obrázku (může obsahovat i cestu k obrázku),
- vystup – cesta pro výstupní soubory,
- convertParams – parametry použité při konvertu obrazového souboru,
- tesseractParams – parametry použité při spuštění programu Tesseract-OCR.

Třída obsahuje čtyři metody, které jsou následně vysvětleny:

- ocr(),
- vykresli(),
- uloz(),
- smaz().

Metoda ocr()

Tato metoda obsahuje dva procesy, které jsou důležité pro anonymizaci obrazových dat. První proces je úprava obrazových souborů a druhý je spuštění Tesseractu za účelem vytvoření hOCR souboru. Jde o metodu s návratovou hodnotou File, tedy vrací zpátky soubor hOCR do třídy Anonymizer.

Metoda vykresli(int X1, int Y1, int X2, int Y2)

Tato metoda vykreslí do grafiky **g** černý obdélník podle souřadnic, které se předají při zavolání metody.

Metoda uloz()

Metoda uloz() slouží jako finální úprava obrazového souboru. V této metodě je vytvořený nový soubor s předponou „anon_“ což značí, že se jedná o již anonymizovaný soubor. K souboru je pak přiřazen upravený obrazový soubor ve formátu PNG.

Metoda smaz()

Tato metoda spustí dva procesy na smazání přebytkových a pomocných souborů. První proces smaže soubor hOCR a druhý proces smaže pomocný obrazový soubor JPG.

4.3.4. Třída Retezec

Třída Retezec obsahuje objekt Retezec, který je složen z následujících proměnných:

- slovo – název řetezce,
- cetnost – četnost výskytu řetězce,

Třída má jednu metodu a to:

- compareTo(Retezec porovnejSlovo)

Metoda compareTo(Retezec porovnejSlovo)

Metoda compareTo() slouží jako porovnávací metoda pro seřazení kolekce výskytů. Seřazuje porovnané řetězce sestupně. Vstupní hodnota je řetězec. Po porovnání četností řetězců se vrací jako výstupní hodnota integer.

4.3.4. Třída KolekceVyskytu

Třída KolekceVyskytu je složena z jedné metody a to:

- add(String slovo)

Metoda add(String slovo)

Metoda add() slouží ke vkládání řetězců do kolekce výskytů. Metoda obsahuje cyklus **for**, který porovnává příchozí řetězec se všemi řetězci v kolekci výskytů. Pokud je nalezen totožný řetězec, zvýší se četnost řetězce o 1. V opačném případě je přidán řetězec do kolekce výskytů s počáteční hodnotou 1.

4.4. Postup při tvorbě programu

V této kapitole budou podrobněji popsány metody, proměnné a také postup při tvorbě anonymizačního programu.

4.4.1. Úprava obrazových souborů

V bakalářské práci byl ImageMagick použit pro změnu rozlišení původního obrazového souboru. Důvodem této změny byla příprava souboru pro práci s programem Tesseract. Zjistil jsem totiž, že na obrazových souborech, které mně byly pro práci přiděleny, není úspěšnost rozeznávání písmen a číslic zdaleka tak uspokojivá. Na oficiálních stránkách Tesseractu jsem se dočetl, že změnou velikosti rozlišení se zlepší i kvalita načítání textu, viz tabulka č. 1. K dosažení svého cíle jsem tedy použil **– resize** z knihovny ImageMagick.

Pomocí příkazu – resize jsem zkoušel zadat několik hodnot a porovnával jsem, z jaké hodnoty dostanu nejlepší výsledek – tedy poměr velikosti souboru a schopnosti Tesseractu rozeznat vyžadované znaky.

Tabulka 1: Porovnání velikostí a úspěšnosti rozeznávání textu pro formát PNG

Rozlišení v pixelech	Velikost	Ukázka textu
Originální (980x980)	500 kB	Fakunm nemacmce men
5000x5000	5 000 kB	Fakuitni nemocnice Plzen
10000x10000	14 000 kB	Fakuitni nemoonioe Plzen

Z výše uvedené tabulky je jednoznačně vidět velký rozdíl mezi originálním obrazovým souborem a mnou upraveným souborem.

Rozdíl mezi rozlišením 5000 a 10000 není už tak razantní, avšak lepší hodnoty jsem dostával z rozlišení 5000. Při rozhodování záleželo také na velikosti souboru a rychlosti jeho zpracování. Rozhodl jsem se tedy každý upravovaný obrazový soubor nejdříve změnit na 5000x5000 pixelů.

Formát PNG však ve velkém rozlišení nabývá též velké velikosti (viz tabulka výše), která zpomalovala chod programu až o několik vteřin. Rozhodl jsem se tedy při změně rozlišení také změnit formát souboru, v mém případě jsem zvolil formát JPG. Dále jsem testoval, zdali změna formátu neovlivní úspěšnost načítání znaků (viz tabulka č. 2).

Tabulka 2: Porovnání velikosti a úspěšnosti rozeznávání textu pro formát JPG

Rozlišení v pixelech	Velikost	Ukázka textu
Originální (980x980)	500 kB	Fakunm nemacmce men
5000x5000	2 000 kB	Fakultni nemocnice Plzen
10000x10000	5 000 kB	Fakuttni nemocnioe Plzen

Výše uvedená tabulka potvrzuje, že se mé obavy ohledně snížení rozpoznatelnosti znaků nepotvrdily, ba naopak se v určitých pasážích textu rozpoznávání znaků dokonce zlepšilo a formát JPG dosahoval poloviční velikosti než formát PNG.

Pro změnu formátu a rozlišení obrazového souboru tedy použiji následující příkaz:

"convert " + vstup + covertParams + " " + vystup + "pred.jpg"

kde proměnná „vstup“ obsahuje název a cestu k původnímu obrazovému souboru, proměnná „covertParams“ obsahuje všechny parametry pro konvertování obrazového souboru (např. „resize 5000“, „colorspace Gray“) a proměnná „vystup“ obsahuje cestu k adresáři, kam lze zapisovat.

Text „pred.jpg“ určuje, jak se bude jmenovat pomocně vytvořený JPG soubor. Tento soubor se pro skončení prvního cyklu anonymizace smaže, takže je možné tento soubor mít pojmenovaný napevno.

Při snaze docílit co nejlepších výsledků v anonymizaci citlivých dat jsem také zkoušel zadat i jiné parametry, než jen zvětšení obrazového souboru a změnu formátu.

Zkoušel jsem například měnit spektrum barev. Pro tento pokus jsem použil „colorspace Gray“ parametr, který mi přepnul barvy do odstínů šedé.

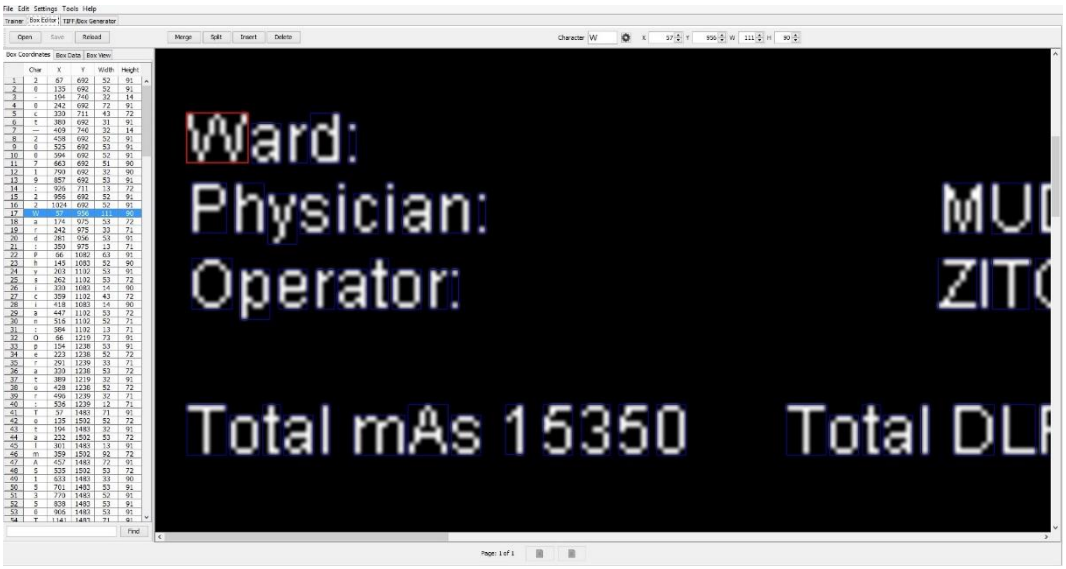
Testováním různých spouštěcích parametrů pro program ImageMagick jsem zjistil, že pro různé soubory je lepší použít různé spouštěcí parametry. Například u souboru, kde je anonymizovaný text na barevném podkladu (konkrétní příklad z dodaných obrazových souborů je modré pozadí s bílým textem), kde většina obrazových souborů obsahovala černý podklad a bílý text.

Zkoušel jsem také změnit i prahování (threshold) obrazového souboru, kdy jsem si vytvořil pomocný jednoduchý podprogram, který mi vygeneroval celkem 40 testových obrazových souborů, každý s jiným procentem prahování. Výsledkem tedy bylo 40 souborů od prahování 30% až do prahování 70%. Tyto soubory jsem pak použil v mém programu jako vstupní obrazové soubory. Bohužel tento způsob se mi neosvědčil, neboť se text stal nečitelným pro program Tesseract-OCR a byl tvořen jen ze znaků „ [“, znaků „]“ a znaků „ | ”.

4.4.2. Trénování Tesseract-Ocr

Tesseract-Ocr umožňuje pro přesnější rozeznávání textu možnost trénování na obrazových souborech. Trénování jsem se rozhodl provádět na zvětšených obrazových souborech (pomocí resize), protože program převádí originální obrazové soubory na rozlišení 5000x5000 (kvůli lepšímu rozeznávání). Po načtení obrazových souborů lze zkontroloval a opravit popř. přidat tzv. boxy obsahující rozeznaný znak. K úpravě jsem použil program jTessBoxEditor.

Obrázek 1: Příklad kontroly boxů v jTessBoxEditor viz Obrázek 2



Po zkontrolování a případné opravě boxů, jsem schopen vytvořit trénovaná data.

Program Tesseract očekává soubory pojmenované podle určitých pravidel. Jedno z těchto pravidel je:

[jazyk].[název fontu].exp[číslo]

Mnou vytvořená trénovací data se mohou například jmenovat takto:

eng1.meddata.exp0.png

eng1.meddata.exp0.box

název eng1 značí, že již existuje jazyk eng.

Pro vytvoření trénování je poté nutné zadat posloupnost příkazů:

tesseract eng1.meddata.exp0.png eng1.meddata.exp0 nobatch box.train

unicharset_extractor eng1.meddata.exp0.box

Je také potřeba definovat přidaný jazyk, tedy:

echo "meddata 0 0 0 0 0" > font_properties

kde první hodnota je název jazyku a další jsou binární hodnoty vlastností fontu – italic, bold, fixed, serif, fraktur.

Po definování jazyku je potřeba vytvořit soubou shapetable, v doslovných překladu tabulku tvarů, ten je však pouze potřeba, jedná-li se o Indické jazyky, proto tento ve trénování přeskočím.

Následují příkazy:

mftraining -F font_properties -U unicharset -O eng1.unicharset

eng1.meddata.exp0.tr

a příkaz:

cntraining eng1.meddata.exp0.tr

Nyní jsem získal všechny potřebné soubory pro vytvoření trénovaných dat v Tesseractu (traineddata), avšak předtím než se vytvoří trénovaná data, je potřeba přidat vytvořeným souborům prefix jazyku. Tuto operaci lze provést ručně nebo pomocí série příkazů:

mv inttemp eng1.inttemp

mv normproto eng1.normproto

mv pffmtable eng1.pffmtable

v případě použití tabulky tvarů také (mv shapetable eng1.shapetable).

Po přidání prefixu zbývá už jen dva poslední kroky. První krok je vytvoření trénovaných dat pomocí příkazu:

combine_tessdata eng1.

a druhým krokem je přesunutí trénovaných dat do příslušného adresáře s daty pro spuštění, např.:

sudo cp eng1.traineddata /usr/local/share/tessdata/

Tesseract-Ocr také umožňuje spuštění s více jazyky najednou:

tesseract obrazek.png output -l eng+eng1

kdy bude použit jazyk eng i jazyk eng1.

4.4.3. Spuštění Tesseract a získání hOCR souboru

Po převedení obrazového souboru je dalším krokem zjištění citlivých dat. Program Tesseract naskenuje obrazový soubor a uloží jej do mnou zvoleného výstupního formátu. Program Tesseract používá jako výchozí výstupní formát typ TXT, ten je však pro mojí práci nedostačující, jelikož neobsahuje pozice nalezeného textu.

Zvolil jsem tedy výstup ve formátu hOCR. Jedná se o soubor typu XHTML.

hOCR soubor jsem získal pomocí příkazu:

"tesseract " + vystup + "pred.jpg " + vystup + nazevObrazku + tesseractParams + " hocr"

kde proměnná „vystup“ značí adresář k upravenému JPG obrazovému souboru. Proměnná „nazevObrazku“ obsahuje název originálního obrazového souboru, který je určen k anonymizaci.

Výstupní soubor je ve formátu hOCR. Soubor obsahuje veškerý nalezený text v upraveném obrazovém souboru a také pozice nalezeného textu.

Příklad celého procesu níže.

tesseract /tmp/prej.jpg /tmp/obraz.png –l eng hocr

Příkaz **convert** i příkaz **tesseract** si program sám volá automaticky při spuštění.

Výstupem jsou dva soubory:

- pred.jpg
- data.hocr

Napevno pojmenovaný je obrazový soubor formátu JPG, soubor hOCR je pojmenovaný vždy stejně jako anonymizový obrazový soubor. Příklad pojmenování:

„FNPL_10101010_10101010.DCM.png.hocr“

Soubory hocr takto pojmenované pak ulehčují podrobnější analýzu anonymizace dat, tj. do jaké míry byl řetězec úspěšně načten a podobně. Uvedené číslo je smyšlené.

Oba soubory jsou ukládány do předem zvoleného adresáře ze souboru properties a po skončení programu jsou automaticky smazány.

4.4.4. Nalezení citlivých dat v obrazových souborech

Po získání hOCR souboru a upraveného obrazového souboru jsem připraven na anonymizaci citlivých dat v původním obrazovém souboru.

Nejdříve je však potřeba dostat načtené údaje uložené v souboru hOCR a abych toho docílil, použil jsem následující metody z knihoven JSoup.

```
org.jsoup.nodes.Document doc = Jsoup.parse(souborHocr, "UTF-8");
```

Tato metoda uložila celý obsah HOCR souboru do dokumentu **doc**, ze kterého pak budu načítat mnou požadovaná data.

Načítání dat probíhá v cyklu **for** a pro uložení získaných řetězců textu jsem vytvořil dvě pole. Do prvního pole budu ukládat nalezené řetězce textu a do druhého pak pozice nalezeného textu.

Parsování pak vypadá takto:

```
for (Element ocrxWord : doc.select(".ocrx_word")) {

    jmena.add(ocrxWord.text()); //Jmeno, Prijmeni

    pozice.add(ocrxWord.attr("title"));  //bbox 250 192
    1606 375; x_wconf 70
```

Cyklus hledá pouze text v souboru hOCR, tedy hodnoty „**.ocrx_word**“.

Do kolekce **jmena** se pomocí příkazu add uloží všechny stringy z dokumentu, které byly naskenovány pomocí Tesseractu.

ocrxWord.attr(“title“); uloží do kolekce všechny hodnoty atributu **title**, mezi nimiž jsou i námi vyžadované pozice načtených stringů.

Každý string je po uložení kontrolován, zdali splňuje předem daná pravidla pro anonymizaci. Pokud splňuje kontrolovaný textový řetězec pravidla, lze tvrdit, že řetězec obsahuje citlivá data, jako je například jméno pacienta, rodné číslo nebo datum narození.

Metoda pro kontrolu textového řetězce se nazývá **zkontrolujString(int i)**. Index **i** odkazuje na hodnotu momentálně testovaného řetězce. Tato metoda je typu **void**, tudíž nemá žádnou návratovou hodnotu.

Metoda zkontrolujString(int i);

Tato metoda, jak již název napovídá, je určena pro kontrolu stringové proměnné. Může se jednat o jméno, příjmení i o datum narozeni.

Pro procházení řetězce jsem použil cyklus **for**, který kontroluje každý char v řetězci, jestli se jedná o písmeno nebo číslo. Jsou zde také uvedeny podmínky, že mezi akceptovatelné znaky patří čárka, tečka a stříška. Důvod uvádím v analýze anonymizace obrazových dat.

Pokud cyklus zjistí, že se jedná o číslo, zavolá se metoda **zkontrolujCislo(int i)** a cyklus se ukončí pomocí break.

Jedná-li se o řetězec složený pouze z písmen abecedy nebo akceptovaných znaků, je nutné splnit další podmínky pro anonymizaci. První podmínka je, zda se jedná o „jméno“ (definuji jako obecný název pro citlivý údaj, může to být i příjmení nebo titul) a zda textový řetězec je větší nebo roven 3 znakům.

Pokud je tato podmínka splněna, textový řetězec se přesune k druhé podmínce. Druhá podmínka je kontrola metodou **porovnejJmeno(int i)**. Tato metoda je relativně jednoduchá, slouží jen k porovnání anonymizovaného textového řetězce s textovým řetězcem ze seznamu bílých slov. Tento seznam prochází s cyklem while, dokud nedojde na konec souboru bilaSlova. Jelikož se jedná metodu typu boolean, jako návratová hodnota této funkce bude true nebo false.

V této metodě jsem musel ošetřit případy, kdy se z Tesseractu načetl řetězec, který obsahoval na konci jména znak tečky nebo čárky viz tabulka č. 3.

Tabulka 3: Příklad ukládání řetězců do pole stringů

Původní text	**První řetězec**	**Druhý řetězec**
PŘIJMENÍ, JMÉNO	PŘIJMENÍ,	JMÉNO

Metoda dokáže akceptovat tečku nebo čárku, vyskytují-li se na konci řetězce a zároveň, pokud je řetězec delší než tři znaky. Je tomu tak z důvodu, aby nebyl akceptován řetězec složený ze dvou čárek jako jméno.

Pokud splňuje nalezený řetězec všechna pravidla, program rozhodne, že je nutné tento řetězec anonymizovat. Je zavolána metoda **anonymizujJmeno(int i)**, která anonymizaci provede. Jak tato metoda funguje a jak probíhá anonymizace, vysvětlím v další části.

Metoda zkontrolujCislo(int i);

Metoda sloužící ke kontrole, zda anonymizovaný textový řetězec je nebo není rodné číslo, datum narození nebo čas pořízení. To se kontroluje metodou **jeCisloZnak(int i, int k)**.

Metoda jeCisloZnak(int i, int k);

Tato metoda kontroluje každý znak anonymizovaného řetězce, zda se jedná o číslo nebo speciální povolený znak (např. otazník, čárka, tečka). Jedná se o metodu typu boolean, takže vrací true nebo false. Po splnění kritérií (je to číslo nebo znak jenž má více než 7 a méně než 12 znaků celkově) pro anonymizaci se zavolá metoda **anonymizujCislo(int i)**.

V metodě jsem ošetřil případy, kdy načtená hodnota z Tesseractu nemusela odpovídat originální hodnotě, viz tabulka č. 4.

Tabulka 4: Příklad nepřesně načteného řetězce z Tesseractu

Původní text	**Načtený text**
12/23/4567	1272374567
123456789	123456?89

Metoda dokáže akceptovat i takto nepovedeně načtené řetězce.

Zavedl jsem pravidlo, že řetězec musí být delší než 7 znaků. Zmíněné pravidlo je zapotřebí, nastane-li podobný případ jako ve výše uvedené tabulce, kdy Tesseract zamění při načítání znak lomítka „/“ za číslici **7**. Z řetězce o 8 číslicích se pak rázem stane řetězec obsahující 10 číslic.

Po nalezení citlivých dat se zavolá funkce **anonymizujCislo()**. Zavolanou funkci vysvětlím v další části.

4.4.5. Anonymizace nalezených citlivých dat

Pro anonymizaci nalezených citlivých dat jsem vytvořil tři metody:

- **anonymizujJmeno();**
- **anonymizujCislo();**

Metody fungují na podobném principu. Jedná se o metody typu void, takže nevrací žádnou návratovou hodnotu.

V každé metodě se zavolá vnořená metoda **zapis();** a vypočte se šířka (width) a výška (height) obdélníku pro vykreslení.

Následně se zavolá metoda **vykresli();**, která použije vypočtenou šířku a výšku a předá jí společně s přepočtenými souřadnicemi anonymizovaného textu.

Metoda zapis();

Abych v programu zamezil zbytečnému opakování stejných příkazů, vytvořil jsem metodu **zapis()**. Metoda obsahuje switch s dvěma case a s default řešením. V programu do této metody posílám 3 stringy:

- neni – posílám do zápisu na konci programu, pokud není nutná anonymizace, posílá se pouze jednou za cyklus,
- nutna – posílám do zápisu, nalezne-li se první údaj, který je potřeba anomizovat posílá se pouze jednou za cyklus,
- „“ – posílám do zápisu, naleznu-li citlivý údaj, který chci zapsat, může být voláno několikrát za jednu iteraci programu.

default (case default)

Obsahuje pole **rozlozeniPozic[]**, do kterého se pomocí funkce split rozdělí řetězec na podřetězce.

Ukázka kódu:

```
rozlozeniPozic = pozice[i].split(" |\\;|\\,|\\^");
```

Tento proces je nezbytný k získání pozic řetězce, neboť původní text obsažený v poli **pozice[]** nemá správnou formu (viz část Nalezení citlivých dat – parsování).

Příklad řetězce pole **pozice[i]**:

bbox 250 192 1606 375; x_wconf 70

Pomocí funkce split tedy získáme hodnoty, viz tabulka č. 5.

Tabulka 5: Rozdělení hodnot pomocí funkce split

k = 0	**k = 1**	**k = 2**	**k = 3**	**k = 4**	**k = 5**	**k = 6**
bbox	250	192	1606	375	x_wconf	70

Kde hodnota **k** značí index pole **rozlozeniPozic[]**.

Pro anonymizaci jsou důležité pouze souřadnice nalezeného textu, tedy **X1**, **Y1**, **X2** a **Y2**. Získaná hodnota z funkce split je textový řetězec, proto je nutné pomocí parseru hodnoty převést na číselné.

Po převedení dosadím hodnoty k=1 až k=4 do proměnných **X1**, **Y1**, **X2** a **Y2**.

Získání souřadnic ovšem není poslední krok, protože tyto souřadnice jsou určené pro upravený obrazový soubor, tedy ten s rozlišením 5000x5000. Dalším krokem je převedení souřadnic pro anonymizaci původního obrazového souboru.

Na to jsem vytvořil metodu **prepocti();**.

Po získání potřebných souřadnic pak zavolám metodu v**ykresli();**

vykresli(int X1, int Y1, int X2, int Y2);

kde **X1**, **Y1**, **X2**, **Y2** jsou pozice nalezeného textu.

V průběhu chodu program vypisuje na obrazovku veškeré úkony, které provádí a zároveň zapisuje do logovacího souboru důležité informace o anonymizaci.

Na začátku logovacího souboru je zapsán obrázek a informace, zda obsahuje citlivá data či nikoliv (Private data/Clean).

Příklad zápisu logovacího souboru:

obrazek.png Private data

obrazek.png 25 19 160 37 JMENO

Jednotlivé údaje jsou odděleny tabulátorem.

Metoda prepocti();

Metoda funguje na jednoduchém principu trojčlenky, kdy vím, že upravený soubor bude vždy v rozlišení 5000x5000, takže stačí jen procentuálně přepočítat souřadnice pro původní obrazový soubor.

Při každé anonymizaci program zapisuje do souboru JSON četnost výskytů anonymizovaných textových řetězců. Tento soubor slouží jako pomůcka při tvorbě seznamu bílých slov. Lze říci, že nejčastěji nalezená slova napříč všem obrazovým souborům budou slova, které není třeba anonymizovat (např. Blood, mAs, atd.). Tohle pravidlo ale neplatí vždy. Na většině obrazových souborů je také poznamenán název nemocnice a i to lze považovat za citlivý údaj, neboť tento údaj může napomoci k odhalení identity pacienta.

5. Zhodnocení výsledků

V této poslední kapitole bych rád zhodnotil dosažené výsledky. Cílem bakalářské práce bylo vytvořit algoritmus zlepšující anonymizaci dat. Vytvořené metody splňují požadavek zvýšené bezpečnosti citlivých údajů při práci s obrazovou částí. Program dokáže identifikovat a odstranit citlivé osobní údaje z obrazových souborů.

Z původních dodaných souborů bylo pro program použitelných pouze 64 z 86, protože zbývajících 22 souborů neobsahovalo žádná data či žádný znak. Při zpracování programu jsem použil několik dostupných aplikací, které jsem uváděl v praktické části a které byly nezbytné pro dosažení optimálních výsledků. Dále byla vytvořena databáze tzv. bílých slov, což jsou slova, která při shodě nalezení v textu se nepovažují za citlivá data. Omezujícím prvkem při tvorbě programu byl OCR program Tesseract verze 3.03, který přes vysokou úspěšnost rozeznávání znaků nebyl perfektní. Tuto vadu jsem se snažil do jisté míry v programu opravit zvětšením obrazového souboru a také pomocí trénování programu Tesseract. Tesseract také rozděloval textové řetězce tam, kde si myslel, že se vyskytuje mezera a začátek nového řetězce. Program byl vyvíjen a testován na platformě Linux Ubuntu 14.04LTS.

Při anonymizování obrazových souborů jsem též zkoušel různá nastavení pro program ImageMagick, který slouží pro úpravu zdrojového obrazového souboru za účelem přesnějšího načítání znaků z programu Tesseract-ORC. Zjistil jsem, že jedno nastavení nelze aplikovat na všechny obrazové soubory, co jsem dostal. Pro některé obrazové soubory postačí základní nastavení, tedy zvětšení obrazového souboru a změna formátu. Obrazové formáty s barevným podkladem pod citlivými daty zase vyžadovaly změnu barev na odstíny šedi.

Také jsem si všiml (po analýze souborů hOCR), že program Tesseract-OCR v některých obrazových souborech vynechává oblasti s citlivými daty. Přes moji snahu se mi tento problém nepovedlo odstranit.

Celkový čas anonymizace všech 86 obrazových souborů se pohyboval kolem 6 a půl minuty. Z toho vyplývá, že na anonymizování jednoho snímku program potřebuje 4,5 vteřin.

Pravidla počítání

Pro získání výsledků jsem musel vytvořit pravidla na počítání anonymizovaných řetězců.

Jméno a příjmení počítám jako jeden údaj stejně tak jako řetězec „Fakultni nemocnice Plzen“. Datum, čas pořízení a jiná čísla (rodné číslo,…) počítám jako jeden řetězec. Tesseract-OCR občas nerozezná jedno slovo jako jeden řetězec, ale rozdělí ho na dva řetězce. Proto za dostatečnou anonymizaci řetězce považuji, když nelze určit ze zbylého řetězce důležitý údaj, podle kterého by mohl být identifikován pacient.

Celkově jsem napočítal podle mnou určených pravidel v obrazových souborech 306 řetězců. Pomocí mého programu jsem byl schopen anonymizovat 246 řetězců, což je přibližně 80% všech řetězců. Z 64 souborů pro anonymizaci byl můj program schopný anonymizovat rovných 20 obrazových souborů (anonymizace 100%, tedy žádná citlivá data), což je přibližně 31%.

CD s přílohou obsahuje všechny anonymizované obrazové soubory. Na těchto obrazových souborech byly použity dvě metody convert - defaultní resize a colorspaceGray. Černou barvou jsou anonymizovány údaje pomocí mého programu a červenou barvou jsou označeny údaje, které jsem anonymizoval ručně.

Výsledkem mé práce je program, který anonymizuje medicínská data, aniž by se musela nahrávat na neznámé webové stránky, které by mohly ohrozit soukromí dat, a tím i následné soukromí pacientů či případných dalších zájmových skupin.

V dalších verzích programu by bylo možné rozšířit funkčnost programu, např. vytvořením učenlivé databáze pro jména a příjmení, která by usnadnila anonymizaci dat. Dále by bylo možné vytvořit grafické rozhraní s možností prohlížení upravených obrazových souborů.

Závěr

Tato bakalářská práce byla zaměřena na analýzu legislativních bezpečnostních požadavků pro zpracování medicínských dat a vytvoření anonymizačního programu pro zpracování citlivých dat obsažených v obrazové části DICOM souborů.

V této práci jsem nejprve osvětlil úvod do bezpečnostní problematiky a poté jsem rozvedl právní předpisy a zákony, které s touto problematikou přímo, či okrajově souvisejí. Zmínil jsem i zahraniční zákon HIPAA a hlavy tohoto zákonu. Nastínil jsem též i předpisy pro nakládání s lékařskými a osobními údaji ve zdravotnickém zařízení.

V dalším bodě teoretické části jsem představil nejpoužívanější formát pro medicínská data, jímž je formát DICOM. Zde jsem uvedl i něco z historie, nejenom v České republice a popsal jeho základní části. Také jsem popsal systémy pro správu a archivaci DICOM souborů, tedy PACS.

Dále jsem uvedl a zhodnotil dostupné metody anonymizace. Většina programů zabývajících se anonymizací DICOM souborů je pro mě nepoužitelná, neboť pracují s metadaty DICOM souborů. S těmi já ve svojí bakalářské práci nepracuji a anonymizuji pouze vyexportovanou obrazovou část formátu PNG ze souborů DICOM. Narazil jsem také na tzv. browserové aplikace na anonymizaci obrazových souborů – převážně fotek (opět nepoužitelné pro anonymizaci). Kromě anonymizace fotek jsem našel i browserový anonymizer ZORRO, který také popisuji. Ačkoliv jsou browserové aplikace užitečné, vyšlo mi z této analýzy najevo, že se nejedná o dostačující řešení pro moji problematiku. Nejlepším řešením tedy zůstává navržení a implementace vlastního programového řešení.

V praktické části jsem se věnoval analýze obrazových souborů, které jsem obdržel na anonymizaci. Podle mnou provedené analýzy jsem vytvořil pravidla pro implementaci anonymizační program. V programu jsem se věnoval vytvoření souborů hOCR pomocí programu Tesseract-Ocr, ze kterých jsem poté mohl citlivá data získávat. Za pomoci knihoven JSoup jsem z těchto souborů mohl postupně parsovat citlivá data. Po vyparsování zjistí mnou vytvořené metody, zda se jedná o data citlivá. Pokud ano, jsou nalezená data okamžitě anonymizována. V opačném

případě program tato data ignoruje. Nalezená data jsou automaticky zapisována do logovacího souboru a do souboru JSON, který obsahuje četnosti jednotlivých řetězců. Průběžný stav programu je vypisován na obrazovce počítače. Po kontrole všech získaných řetězců ze souboru hOCR se všechny pomocné soubory, vytvořené pro anonymizaci smažou. Mezi tyto pomocné soubory patří hOCR a pomocný zvětšený obrazový soubor formátu JPG.

Jsem vděčný za možnost zpracování tohoto tématu, a to z důvodu, že mně bylo umožněno nahlédnout i do fungování zdravotnictví. Jelikož velká část materiálů, které jsem obdržel či si vyhledával, byly v anglickém jazyce, měl jsem také možnost si prověřit své jazykové znalosti a mohl jsem aplikovat dosažené teoretické znalosti v oboru Informační systémy.

Literatura a prameny

1. **Konference ICT ve zdravotnictví | Inflow.** ***Inflow | magazín nejen pro knihovníky.*** **[Online] [Citace: 24. duben 2017.] http://www.inflow.cz/konference-ict-ve-zdravotnictvi.**

2. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění.*** **[Online] [Citace: 24. duben 2017.] http://www.zakonyprolidi.cz/.**

3. Kybernetický zákon. ***Kybernetický zákon.*** **[Online] [Citace: 24. duben 2017.] http://kybernetickyzakon.cz/.**

4. WWW.CLK.CZ: Úmluva o lidských právech a biomedicíně:. ***WWW.CLK.CZ: Česká lékařská komora - OS Děčín (Index):.*** **[Online] [Citace: 24. duben 2017.] http://www.clk.cz/oldweb/zakpred/Uml096-2001_EtikaBiomed.html.**

5. Informace o NZIS | ÚZIS ČR. ***ÚZIS ČR | Ústav zdravotnických informací a statistiky ČR.*** **[Online] [Citace: 24. duben 2017.] http://www.uzis.cz/nas/informace-nzis.**

6. Zákon o zdravotních službách - č. 372/2011 Sb. - Aktuální znění:. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění:.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2011-372.**

7. Vyhláška o zdravotnické dokumentaci - č. 98/2012 Sb. - Aktuální znění:. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění:.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2012-98.**

8. Zákon o ochraně osobních údajů - č. 101/2000 Sb. - Aktuální znění. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2000-101.**

9. Zákon o kybernetické bezpečnosti a o změně souvisejících zákonů (zákon o kybernetické bezpečnosti) - č. 181/2014 Sb. - Aktuální znění:. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění:.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2014-181.**

10. Předpis č. 316/2014 Sb. ***https://www.zakonyprolidi.cz.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2014-316.**

11. Sb., Předpis č. 317/2014. Vyhláška o významných informačních systémech a jejich určujících kritériích - č. 317/2014 Sb. - Aktuální znění:. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2014-316.**

12. What is HIPAA (Health Insurance Portability and Accountability Act)? . ***Health IT and Electronic Health information, news and tips - SearchHealthIT.*** **[Online] [Citace: 24. duben 2017.] http://searchhealthit.techtarget.com/definition/HIPAA.**

13. Guidance for research DICOM images - Guidance for research DICOM images.pdf. ***Duke University School of Medicine.*** **[Online] [Citace: 24. duben 2017.] https://medschool.duke.edu/sites/medschool.duke.edu/files/field/attachments/Guidance%20for%20research%20DICOM%20images.pdf.**

14. DICOM: About DICOM. ***DICOM Homepage.*** **[Online] [Citace: 24. duben 2017.] http://medical.nema.org/Dicom/about-DICOM.html.**

15. DICOM Homepage. ***DICOM Homepage.*** **[Online] [Citace: 24. duben 2017.] http://medical.nema.org/standard.html.**

16. What is PACS (picture archiving and communication system)? ***Health IT and Electronic Health information, news and tips - SearchHealthIT.*** **[Online] [Citace: 24. duben 2017.] http://searchhealthit.techtarget.com/definition/picture-archiving-and-communication-system-PACS.**

17. DICOM Anonymizer Pro. ***NeoLogica: software solutions for medical imaging.*** **[Online] [Citace: 24. duben 2017.] https://www.neologica.it/html/products/DICOMAnonymizerPro.**

18. Drag 'n Drop batch Anonymization of DICOM data for free - doRadiology.com. ***Drag 'n Drop batch Anonymization of DICOM data for free -***

doRadiology.com. **[Online] [Citace: 24. duben 2017.] https://dicomanonymizer.com/index.html.**

19. DICOM Anonymizer download | SourceForge.net. ***DICOM Anonymizer download | SourceForge.net.*** **[Online] 24. duben 2017. https://sourceforge.net/projects/dicomanonymizer/.**

20. Replace, empty, and/or remove patient's, physician's, and any other information from DICOM files - DICOM Anonymizer. ***DICOM Apps: DICOM Converter | DICOM Anonymizer | DICOM to JPEG | JPEG to DICOM | DICOM to NIfTI | NIfTI to DICOM | DICOM to GIF | DICOM to Video | DICOM Thumbnailer.*** **[Online] [Citace: 24. duben 2017.] http://www.dicomapps.com/dicom-anonymizer/index.html.**

21. Anonymize IJ DICOM. ***ImageJ.*** **[Online] 24. duben 2017. https://imagej.nih.gov/ij/plugins/anonymize-ij-dicom/index.html.**

22. Conquest DICOM software. ***Conquest DICOM software.*** **[Online] [Citace: 24. duben 2017.] https://ingenium.home.xs4all.nl/dicom.html.**

23. ImageJ. ***DICOM Rewriter.*** **[Online] [Citace: 24. duben 2017.] https://imagej.nih.gov/ij/plugins/dicom-rewriter.html.**

24. LONI Inspector | Laboratory of Neuro Imaging. ***LONI: Laboratory of Neuro Imaging.*** **[Online] [Citace: 24. duben 2017.] http://www.loni.usc.edu/Software/LONI-Inspector.**

25. DicomEditor - MircWiki. ***MIRC Overview - CTP and TFS - MircWiki.*** **[Online] [Citace: 24. duben 2017.] http://mircwiki.rsna.org/index.php?title=DicomEditor.**

26. ATBON. Anonymizace a úprava dokumentů. [Online] Atbon, a.s. [Citace: 24. duben 2017.] http://www.atbon.cz/redacting.ph.

27. ***tesseract-ocr An OCR Engine that was developed at HP Labs between 1985 and 1995... and now at Google. - Google Project Hosting.*** **[Online] [Citace: 24. duben 2017.] https://github.com/tesseract-ocr/.**

28. ImageMagick: Convert, Edit, Or Compose Bitmap Images. ***ImageMagick: Convert, Edit, Or Compose Bitmap Images.*** **[Online] [Citace: 24. duben 2017.] http://www.imagemagick.org/script/index.php.**

29. jsoup Java HTML Parser, with best of DOM, CSS, and jquery. ***jsoup Java HTML Parser, with best of DOM, CSS, and jquery.*** **[Online] [Citace: 24. duben 2017.] http://jsoup.org/.**

30. VietOCR. Tesseract box editor & trainer. ***VietOCR.*** **[Online] [Citace: 24. duben 2017.] http://vietocr.sourceforge.net/training.html.**

31. O projektu. ***ePACS - DICOM komunikace mezi zdravotnickými zařízeními.*** **[Online] [Citace: 24. duben 2017.] http://www.epacs.cz/faces/pages/o-projektu.xhtml.**

32. Zákon o zdravotních službách - č. 372/2011 Sb. - Aktuální znění:. ***Zákony pro lidi - Sbírka zákonů ČR v aktuálním konsolidovaném znění:.*** **[Online] [Citace: 24. duben 2017.] https://www.zakonyprolidi.cz/cs/2011-372.**

33. Facepixelizer | Pixelate - Blur - Anonymize | Free Online Image Editor:. [Online] [Citace: 24. duben 2017.] http://www.facepixelizer.com/.

34. Lionytics™. Lionytics Image Anonymizer. ***Lionytics Image Anonymizer.*** **[Online] [Citace: 24. duben 2017.] http://www.lionytics.com/lionytics/image-anonymizer/.**

35. PhotoHide. PhotoHide.com - hide the face on your personal photos to ensure your privacy:. ***PhotoHide.com.*** **[Online] [Citace: 24. duben 2017.] http://www.photohide.com/.**

Seznam zkratek

DICOM ..... Digital Imaging and Communications in Medicine

OCR ..... Optical Character Recognition

CT ..... Computed Tomography

ACR ..... American College of Radiolog

NEMA..... National Electrical Manufacturers Association

PACS ..... Picture Archiving and Communication System

HTML ..... HyperText Markup Language

XHTML ..... eXtensible HyperText Markup Language

HOCR ..... Optical Character Recognition

CERT ..... Computer Emergency Response Team

UNLV ..... University of Nevada-Las Vegas

MRI..... Magnetic Resonance Imaging

ÚOOÚ..... Úřad pro ochranu osobních údajů

VNA..... Vendor Neutral Archive

JSON..... JavaScript Object Notation

HIPAA ..... Health Insurance Portability and Accountability Act

PHI..... Protected Health Information

ePHI ..... electronic Protected Health Information

Seznam tabulek

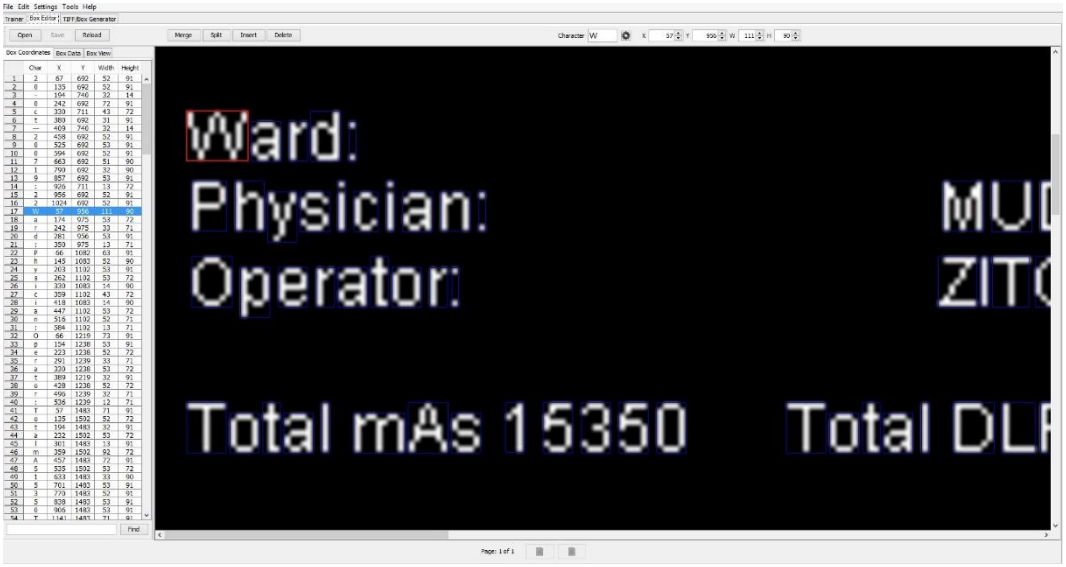
Seznam obrázků

Seznam příloh

- CD obsahující tuto dokumentaci ve formátu PDF a XDOC, zdrojové soubory, podpůrné knihovny a anonymizované obrazové soubory.
- Příloha A
- Příloha B

Příloha A – Upravování boxů v jTessBoxEditor

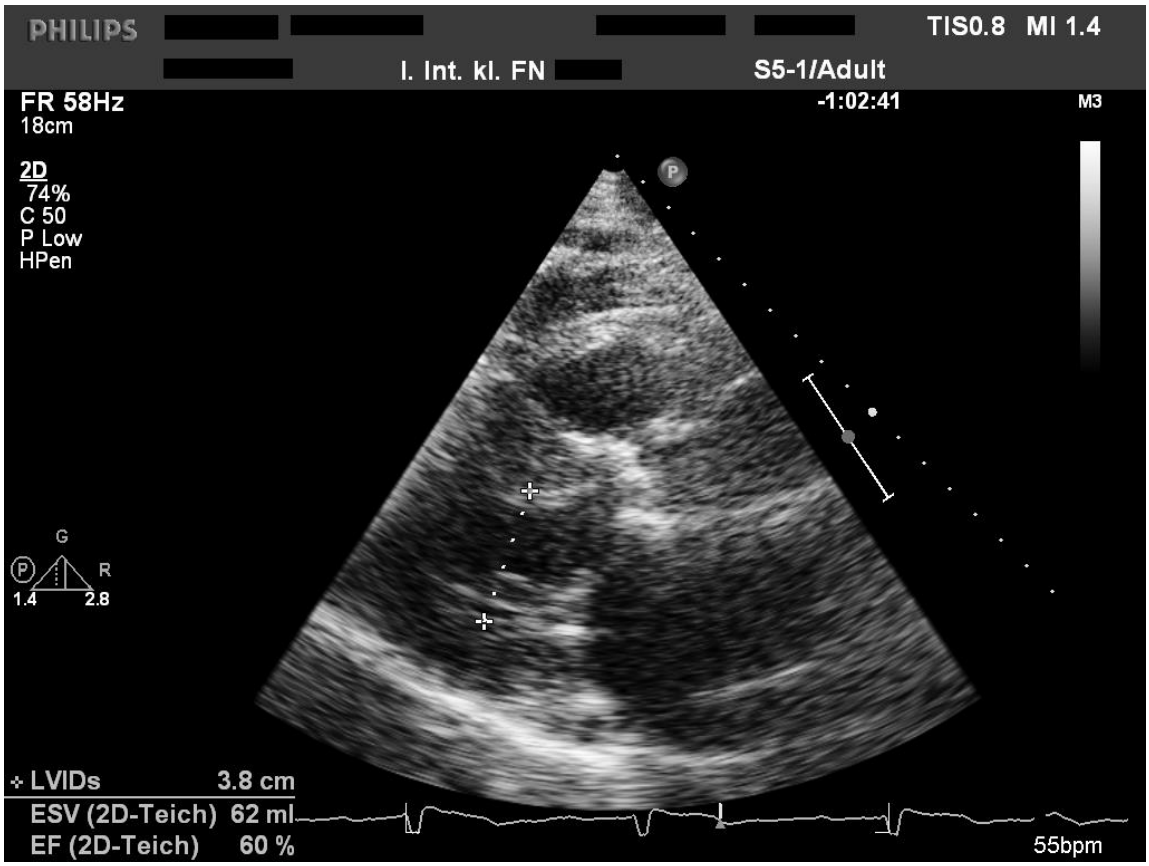
Obrázek 2: Upravování boxů v jTessBoxEditor



Zdroj vlastní.

Příloha B – Anonymizovaná medicínská data

Obrázek 3: Příklad čistého anonymizovaného obrazového souboru



Zdroj vlastní.