VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

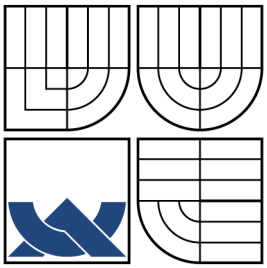
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

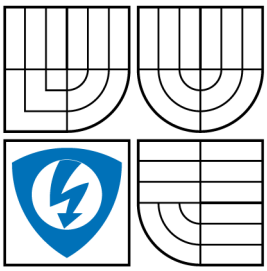
DATOVÁ KOMUNIKACE MEZI ZDRAVOTNICKÝMI INFORMAČNÍMI SYSTÉMY

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE KATEŘINA KUBÍNOVÁ
AUTHOR



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF BIOMEDICAL ENGINEERING

DATOVÁ KOMUNIKACE MEZI ZDRAVOTNICKÝMI INFORMAČNÍMI SYSTÉMY

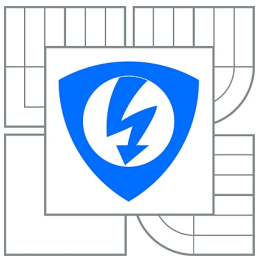
DATA COMMUNICATION BETWEEN THE HEALTH INFORMATION SYSTEMS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — KATEŘINA KUBÍNOVÁ
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ SEKORA
SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav biomedicínského inženýrství

Bakalářská práce

bakalářský studijní obor
Biomedicínská technika a bioinformatika

Studentka:	Kateřina Kubínová	***ID:***	136478
Ročník:	3	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Datová komunikace mezi zdravotnickými informačními systémy

POKYNY PRO VYPRACOVÁNÍ:

1) Proveďte literární rešerši datových standardů ve zdravotnictví a porovnejte zahraniční a národní datový standard Ministerstva zdravotnictví České republiky. 2) Proveďte analýzu zabezpečení současné datové elektronické komunikace mezi různými zdravotnickými organizacemi v síti Internet. Diskutujte možnosti zabezpečení datové komunikace. 3) Na základě provedené analýzy realizujte kryptografický systém pro zabezpečení přenášených dat v podobě uživatelské aplikace server – klient, případně klient – klient. Systém bude v jazyce Java nebo C. 4) Vytvořte model datové zprávy ve standardu MZ ČR, tuto šifrujte navrženým systémem, dešifrujte a vyhodnoťte výsledek přenosu.

DOPORUČENÁ LITERATURA:

[1] World Health Organization. Mezinárodní klasifikace funkčních schopností, disability a zdraví: MKF. Praha: Grada Publishing, 2008. ISBN 978-80-247-1587-2.
[2] SINGH, Simon. Kniha kódů a šifer. Praha: Dokořán, 2003, 382 s. ISBN 80-865-6918-7.

Termín zadání:	11.2.2013	***Termín odevzdání:***	31.5.2013

Vedoucí práce: Ing. Jiří Sekora
Konzultanti bakalářské práce:

prof. Ing. Ivo Provazník, Ph.D.
Předseda oborové rady

UPOZORNĚNÍ:

ABSTRAKT
Bakalářská práce se zabývá tématy datových standardů ve zdravotnictví, národního zdravotního informačního systému a národních zdravotních registrů. Součástí práce je analýza zabezpečení citlivých pacientských dat při komunikaci mezi nemocničními informačními systémy v síti internet.

KLÍČOVÁ SLOVA
Datové standardy, mezinárodní klasifikace nemocí, mezinárodní klasifikace funkčních schopností, disability a zdraví,národní zdravotní informační systém, kryptografie, infrastruktura veřejných klíčů

ABSTRACT
Semestral thesis is applyed to theme of data standards, national information system and national health registers. Main part is analysis of security of sensitive patient´s data during transfer between hospital information systems in network of Internet.

KEYWORDS
Data standards, International classification of dissases, International classification of functioning, disability and health, National health information system, cryptography, Public key infrastructure

KUBÍNOVÁ, Kateřina *Datová komunikace mezi zdravotnickými informačními systémy*: bakalářská práce. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav biomedicínského inženýrství, 2013. 45 s. Vedoucí práce byl Ing. Jiří Sekora

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Datová komunikace mezi zdravotnickými informačními systémy“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

(podpis autora)

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu práce, Ing.Jiřímu Sekorovi, za dodávání optimismu, dobré rady a za odvedenou pečlivou práci při vedení bakalářské práce.

Brno ....

(podpis autora)

OBSAH

SEZNAM OBRÁZKŮ

ÚVOD

Tato práce se věnuje oblasti standardizace dat ve zdravotnictví a zabezpečení datové komunikace mezi nemocničními informačními systémy. 1 První kapitola patří standardům struktury pro přenos informace mezi nemocničními informačními systémy a registry povinných hlášení národního zdravotního informačního systému. Reprezentace obsahu zdravotnického záznamu představuje způsob kódování pro zkrácení administrativy a snazší práci s daty z pacientského záznamu při správě dat v nemocničním informačním systému. V práci jsou rozebrány způsoby zápisu zdravotního stavu jedince podle nově nastupující mezinárodní klasifikace funkčních schopností a disability i podle tradiční mezinárodní klasifikace nemocí. Pár slov o národním zdravotním informačním systému je v kapitole číslo tři 3.

Problematika odesílání povinných hlášení do registrů národního zdravotního informačního systému se přesunula na Internet. Jelikož se jedná o velice citlivá osobní data, je potřeba odesílaná data zabezpečit proti odposlechu nebo jejich poškození od třetí osoby. Komunikace mezi nemocničními informačními systémy a národním zdravotním informačním systémem je uskutečňována v zabezpečené šifrované podobě. V práci je rozebírána problematika symetrické i nesymetrické kryptografie a problematika distribuce klíčů.

Pro praktické ověření teoretických znalostí bylo navrženo sestavení přehledného kryptografického modelu pro šifrování datové zprávy určené k přenosu do národního zdravotního informačního systému.

1 DATOVÉ A KOMUNIKAČNÍ STANDARDY VE ZDRAVOTNICTVÍ

1.1 Český datový standard DS

Specifickou formou datového standardu je náš národní standard. Slouží především k předávání informací mezi jednotlivými informačními systémy. Prošel si složitým vývojem od formátu DS 1.0 (zvaného DaSta) k dnešnímu DS 4. Počátky standardu patřily hledání inspirace v zahraničí. Patrně ale našim potřebám žádný nevyhovoval a tak se MZ ČR rozhodlo dát vlastní cestou. První číselník spatřil světlo světa v roce 1994. Jeho chudá struktura ještě neobsahovala podporu pro laboratorní komplement. V roce 1997 přišla dokonalejší verze DS 1.1 a byla rozšířena o Národní číselník laboratorních pomůcek (NČLP) pro podporu onoho chybějícího komplementu. Struktura však ale nebyla dokonalá a výstup byl nepřehledný textový soubor "*.txt". Verze 2 a 3 už používají soubory " *.DTD", které jsou ovlivněny strukturou XML. Verze DS 3.01 byla rozšířena o číselníky. Dnes aktuální verze DS 4.0 zavádí základní jednotku klinické události s jejich jasnou identifikací. [2]

1.2 Zahraniční standard HL7

Health Level Seven je celosvětově nejrozšířenější datový standard. O jeho propracovanosti svědčí schopnost domluvit se s téměř každým zdravotnickým zařízením na světě. Byl vyvinut v USA a je oficiálním ANSI standardem. Původ tohoto datového standardu se datuje až do roku 1987, kdy byla ustanovena organizace Health Level Seven Inc. a od roku 1994 je titulována jako standardizační organizace.
Standard HL7 zahrnuje ve své struktuře obory jako administrace pacientů, dotazy, žádanky, výsledky vyšetření, grafické výstupy, plánování, průběh péče, správu záznamů, doporučení pacientovi, finanční management a základní datové soubory. Data jsou definována jako předdefinovaná tabulka nebo uživatelsky upravená tabulka. Otevřená architektura takovéto zprávy zaručuje schopnost komunikace s různými systémy. Ve verzi 2 se jedná především o workflow hospitalizační péče. Následná verze 3 si již klade vyšší požadavky na propojení všech procesů probíhajících v nemocnici. Další změnou je snaha o testovatelný standard, který ve druhé verzi není dostupný.[2]

1.3 Standard digitálních obrazových informací DICOM

DICOM (Digital Image and Communications In Medicine protocol suite) je specializovaný datový standard, který vznikl jako odpověď na potřebu všech zobrazovacích metod svá zjištěná data reprezentovat. Lze jej užít jak u 2D signálů(EKG,EEG...) tak i obrazových dat z RTG, CT, PET a pod. DICOM Standard specifikuje síťový protokol TCP/IP, definuje třídy služeb spojených s přenosem dat a přiřazuje unikátní názvy. DICOM je standardem nejen pro obrazová pacientská data, ale také pro zprávy, studie a další data. DICOM počítá s uplatněním ve všech systémech elektronického pacientského záznamu. DICOM je považován za jediný živý standard, který je průběžně upravován, doplňován a aktualizován. Nová verze vychází obvykle jednou ročně.[2]

2 REPREZENTACE OBSAHU ZDRAVOTNICKÉHO ZÁZNAMU

Pro statistické sjednocení zpráv všech lékařů v dané zeměpisné oblasti se používá standardního kódování. Většině onemocnění je přisouzen specifický kód a je tedy daleko jednodušší s tímto pracovat jako se statistickou jednotkou. První lokální nomenklatury vznikaly již v 18.století, aby sloužily k statistickému vyhodnocování příčin úmrtí. V dalším vývoji si světová organizace WHO (World Health Organisation) prosadila dvě varianty číslování nálezů. Je to vhodné pro mezinárodní spolupráci lékařů, která je díky telemedicíně dnes poměrně častější. První číselník, pocházející z roku 1900, se věnuje výhradně číslování nemocí a poruch. Nejnovější verze číselníku, zvaná Mezinárodní klasifikace funkční schopností, disability a zdraví, už zavádí pohled z druhé strany, tedy jaký je životní standard pacienta a co všechno je schopen pro sebe nebo své okolí vykonávat. [2][1]

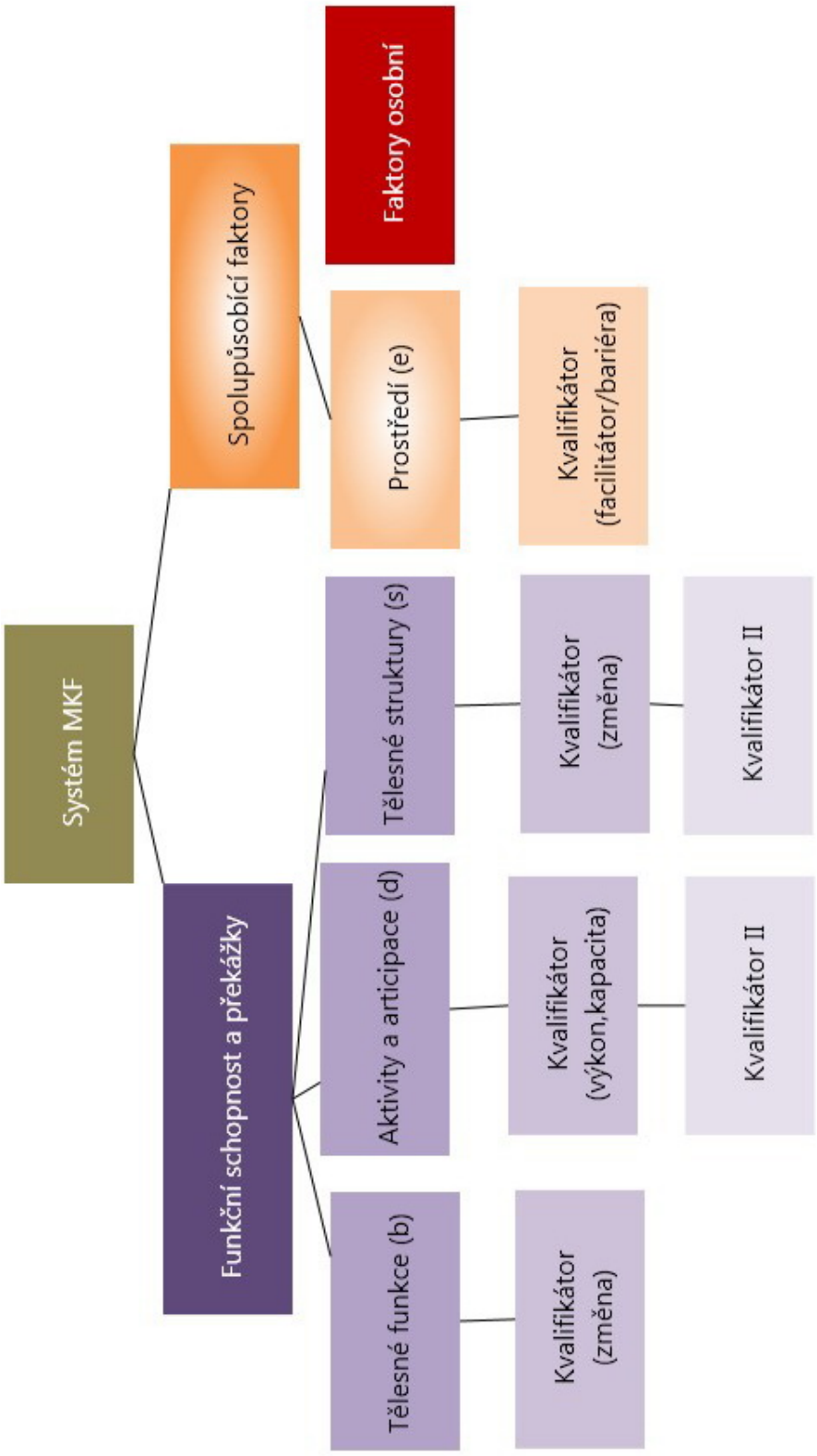
2.1 Mezinárodní klasifikace nemocí

Sborník kódů pro mezinárodní klasifikaci nemocí (MKN) slouží jako pomůcka při zadávání zjištěné diagnózy do nemocničního informačního systému(NIS).Diagnóza je převedena na alfanumerický kód, který usnadňuje vyhledávání a analýzu dat. Kód v poslední, desáté, verzi číselníku MKN má podobu čtyřmístného kódu, kde na prvním místě stojí písmeno, další dvě místa označují specifickou nemoc a čtvrtá pozice za tečkou specifickou podobu onemocnění. Poslední verze MKN stále nezná současně popularizované choroby jako mentální anorexie nebo bulimie, tudíž už by si opět zasloužila novější verzi.[5]
První oficiálně uznaná verze číselníku MKN pochází z mnohaleté práce týmu kolem Jacquese Bertillona. První předložení se konalo roku 1893 v Chicagu. S kladným ohlasem byla publikace přijata. Klasifikace příčin úmrtí (předchůdce dnešní MKN) obsahovala souhrn klasifikací tehdejších evropských lídrů (anglické, německé a švýcarské klasifikace). Sloužila především ke statistickým účelům. Již v roce 1900 na výroční schůzi v Paříži byla přijata nová verze, která již zahrnovala i nemoci nevedoucí ke smrti. Druhá revize za deset let dostala název Mezinárodní klasifikace příčin nemocí a smrti. Ve čtyřicátých letech se, díky všudypřítomnému zmatku, dala spousta zemí cestou národních klasifikací založených na předchozí verzi mezinárodní klasifikace. Rok 1948 definitivně rozhodl pro nový číselník pojmenovaný Příručka mezinárodní statistické klasifikace nemocí, úrazů a příčin smrti. Příručka se skládala ze dvou dílů. První byl seznam kódů a druhý obsahoval abecední seznam položek zahrnutých v jednotlivých kódech. Šestá revizní konference znamenala pro číselníky

novou dobu. Byla sjednocena pravidla pro výběr základní příčiny smrti. Vznikla doporučení pro tvorbu národních oddělení zabývajících se jednotlivými číselníky a statistikou, mezinárodně zpracovatelnou pro světovou zdravotnickou organizaci (WHO). V dnešních dnech je MKN univerzálně používaným číselníkem obsahujícím velké spousty položek a dodatkových číselníků. Je otázkou času či bude nahrazen nově zaváděnou klasifikací funkční schopnosti nebo vznikne jedenáctá revize.[2]

2.2 Mezinárodní klasifikace funkční schopnosti a disability

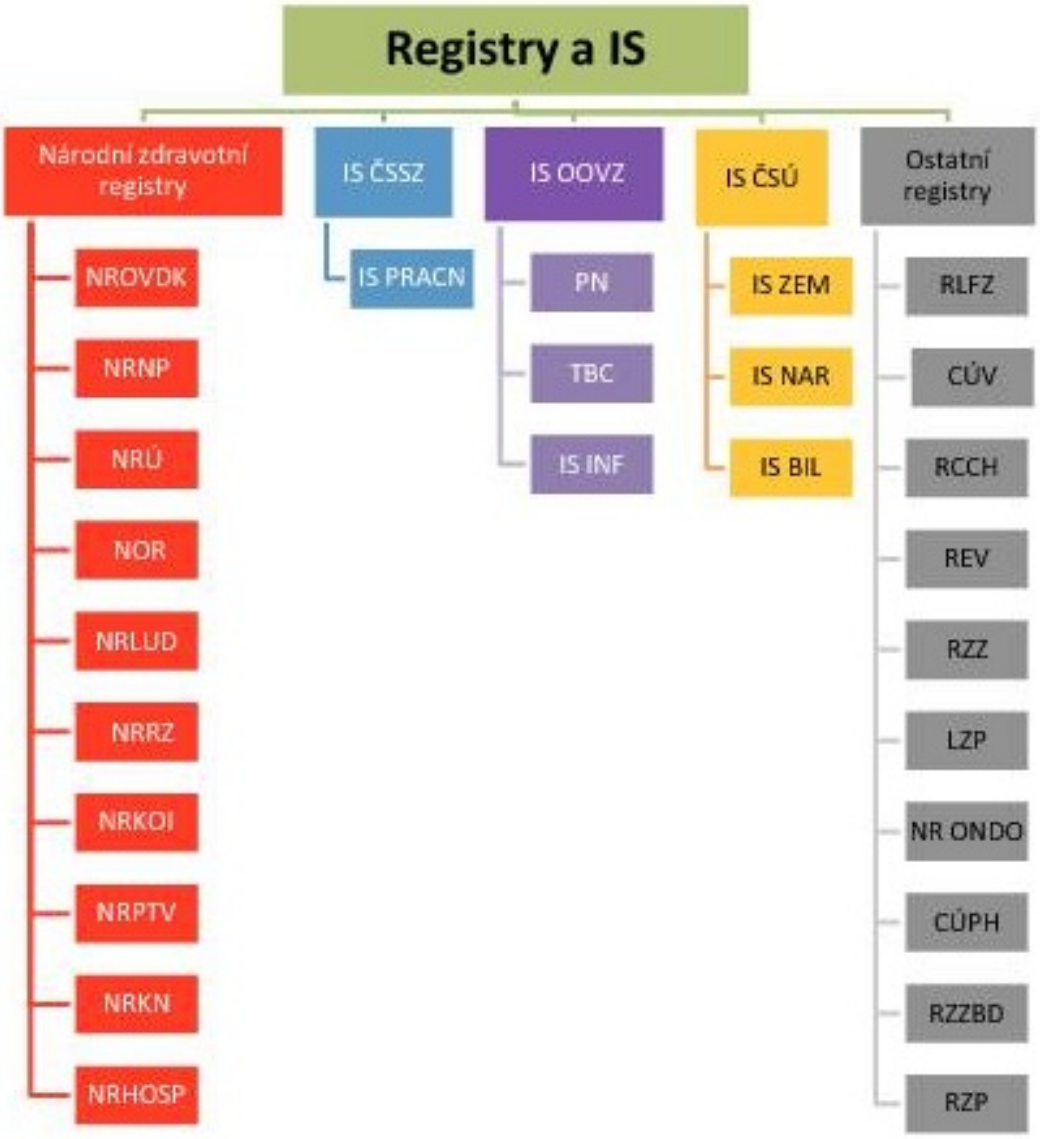
Novým standardem pro klasifikaci se stávají kódy z nomenklatury Mezinárodní klasifikace funkční schopnosti,disability a zdraví (MKF). MKF se nezabývá určováním jednotlivé diagnózy, ale klasifikuje jaké schopnosti pacientovi zůstanou po nemoci nebo úrazu. Hodnocení v MKF je plně využitelné pro nejrůznější posudkové komise. Hodnotí jak dané postižení tak i jeho "sílu". MKF postihuje okamžitý stav v momentu zaznamenání. V žádném případě se nemůže jednat o vyjadřování dalších prognóz nebo následné kvality života. Je dobré hodnocení poměrně často aktualizovat. Následující diagram zobrazuje rozdělení MKF na jednotlivé podkategorie. Písmeno v závorce u názvu kategorie je první předponou kódu. Kvalifikátory obsahují číselnou hodnotu kódu. Celek tvoří pěti až šesti místný kód, kde první kvantifikátor hodnotí obecně nastalý problém a druhý kvantifikátor(po oddělení tečkou) stupeň nastalého postižení.[1]



Obr. 2.1: Systém MKF: Stromový diagram

3 NÁRODNÍ ZDRAVOTNÍ INFORMAČNÍ SYSTÉM

Národní zdravotní informační systém (NZIS) je určen ke sběru zdravotnických dat z celé České republiky. Jednotný zdravotnický systém zajišťuje vedení národních zdravotních registrů. Data získaná z registru je možné využívat v lékařském výzkumu, při tvorbě zdravotní politiky, slouží k ukládání informací o finančních tocích v medicínské péči a k tvorbě statistických přehledů. Vedoucí institucí zodpovědnou za vedení NZIS je Ústav zdravotnických informací a statistiky (UZIS), který spadá svou působností pod ministerstvo zdravotnictví. Pro nadnárodní propojení všech důležitých dat slouží IS WHO, který spravuje organizace WHO a přispívá do něj svými daty vysoký počet ze všech 190 členských států.[2]
Kromě vedení všech národních zdravotních registrů s povinným hlášením je funkcí NZIS i správa registrů nepovinných hlášení. Mezi tyto registry řadíme Národní registr osob nesouhlasících s posmrtným odběrem tkání a orgánů, Registr lékařů, zubních lékařů a farmaceutů, Registr cévní chirurgie, Centrální úložiště povinných hlášení, Centrální úložiště výkazů, Registr ekonomických výkazů, Registr zdravotnických pracovníků způsobilých k výkonu zdravotnického povolání bez odborného dohledu, Registr zdravotnických prostředků, Registr zdravotnických zařízení, Vyplňování listu o prohlídce zemřelého. [3]
Dalšími podčástmi NZIS jsou IS orgánů ochrany veřejného zdraví, IS z datových souborů ČSÚ a IS z datových souborů ČSSZ. Přehlednou strukturu všech registrů národního informačního systému dodává diagram 3.



Obr. 3.1: Registry a IS: Organizační diagram

4 ANALÝZA ZABEZPEČENÍ DATOVÉ KOMUNIKACE V SÍTI INTERNET

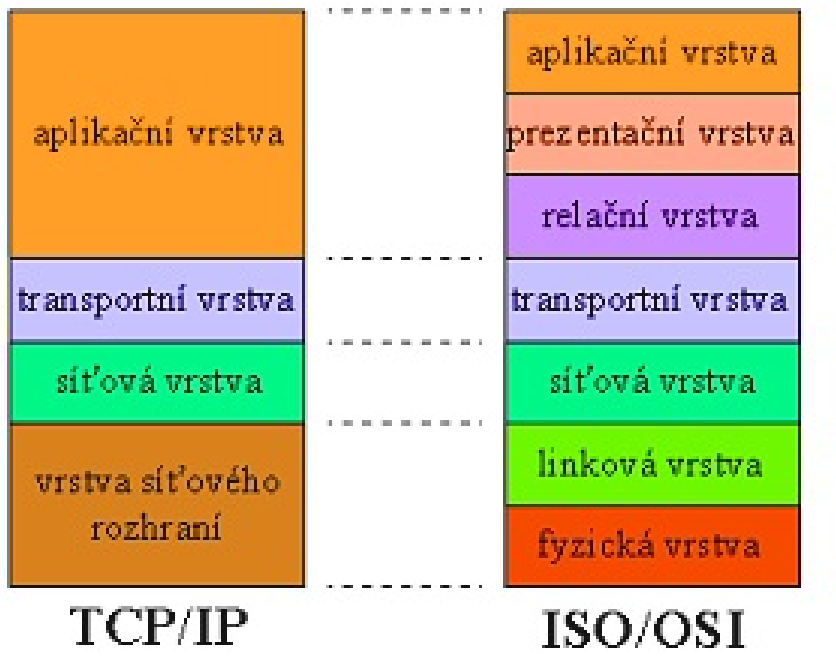
Důležitou součástí všech nemocničních systému jsou komunikační kanály. Ty dovolují NIS získávat aktuální data o pacientech z různých oddělení či zařízení v síti. Jelikož se jedná o citlivá zdravotnická data, podléhajících zákonům na jejich ochranu, musí být transportována zabezpečenými kanály. Prvním stupněm ochrany je zabezpečení komunikace v rámci vlastní intranetové sítě a jejím oddělením od Internetu. Na druhé úrovni je třeba zabezpečit komunikaci, která probíhá v rámci otevřené sítě Internet. Jelikož Internet masivně používá protokol TCP/IP, při jehož návrhu byla cílem spíše efektivnost přenosu než zabezpečení dat, je potřeba vytvářet zabezpečené kanály. Ty zaručují, že po cestě nedojde ke ztrátě nebo poškození dat, také že se data nedostanou k neautorizovanému uživateli. V neposlední řadě není dobré opomíjet fyzické zabezpečení v rámci intranetové sítě. Volíme zabezpečení každé z pracovních stanic heslem pro přístup a bezpečné uzamčení serverového místnosti či prostoru. Jednotlivé typy elektronického zabezpečení se nachází v následujících částech kapitoly.[13]

4.1 Zabezpečení na straně serveru

Server i klient by měli vědět, s kým se hodlají bavit. Proces ověření identity serveru i klienta se nazývá autentizace. Klient může být vyzván k zadání osobních údajů, které potvrdí heslem. Je to běžné například při přihlašování na e-mailový server nebo do kterékoliv jiné zabezpečené aplikace. Bohužel i data jako je přihlašovací jméno a heslo odchází z počítače zcela nezabezpečená. A to nevyhovuje nejpřísnějším kritériím pro bezpečnost zdravotnických dat. O ověření identity pro významnější klientelu se starají certifikační autority. Patří mezi ně významné obchodní firmy jako VeriSign nebo v ČR I.CA. Certifikační autority udělují digitální certifikáty. Digitální certifikát je podepsaný veřejný šifrovací klíč. Funguje na základě přenosu důvěry. Pokud klient věří certifikační autoritě, která certifikát vydala, nemá jistě pochyby o tom, že si ověřila identitu serveru a udělila jej jen vhodnému kandidátovi. Certifikát pro SSL obsahuje sériové číslo, identifikaci majitele certifikátu, údaje o algoritmu použitém k vytvoření podpisu, identifikaci vydavatele certifikátu, datum od kdy do kdy je certifikát platný, obsahuje také účel veřejného klíče (šifrování, ověřování podpisů nebo obojí) a veřejný klíč, dále algoritmus otisku certifikátu a vlastní otisk certifikátu.

4.2 Způsob zabezpečení dat při komunikaci mezi serverem a klientem

4.2.1 Protokol TCP/IP versus model ISO/OSI



Obr. 4.1: Model ISO/OSI versus TCP/IP

Uvedený obrázek 4.2.1 popisuje vrstvy idealizovaného protokolu ISO/OSI (International Standards Organization / Open System Interconnection). Základní rozdíl mezi ISO/OSI a TCP/IP(Transmission Control Protocol/ Internet Protocol) spočívá v počtu vrstev každého modelu. Model ISO/OSI popisuje sedmivrstvý systém, zatímco rodina protokolů TCP/IP má vrstvy pouze čtyři. Základní filozofie ISO/OSI již zahrnuje zabezpečení dat. V protokolu ISO/OSI, v pořadí od nejnižší vrstvy, zastává čestnou první pozici fyzická vrstva. Fyzická vrstva představuje základ pro komunikaci. Pracují na ní veškeré huby a síťové adaptéry. Hlavní funkcí je navazování a ukončování spojení s komunikačním médiem a funkce A/D i D/A převodníku. Druhou vrstvou je vrstva linková, která poskytuje spojení mezi dvěma systémy. Protokoly této vrstvy dělíme na řízený přístup k médiu(Medium Acces Control-MAC) a logicky řízené jako například Ethernetový protokol IEEE 802.2. Z fyzického vybavení se zde vyskytují mosty a přepínače. Síťová vrstva umožňuje komunikaci i s jiným než sousedním systémem. Čtvrtá vrstva tvoří transportní kanál

mezi dvěma systémy. Obsahuje transportně (TCP) a netransportně (UDP) orientované protokoly. Pátá vrstva je relační. Zajišťuje synchronizaci komunikace z obou stran. Je potřebná k zabezpečení, podporuje správní funkce. Předposlední, prezentační vrstva, překládá data do podoby srozumitelné pro aplikaci. V aplikační vrstvě již probíhají protokoly umožňující komunikaci. Například přenos souborů pomocí protokolu FTP, nebo čtení pošty na serveru POP3.
Oproti tomu u skupiny protokolů TCP/IP je pouze vrstva síťového rozhraní, síťová, transportní a aplikační. Základní model TCP/IP spěje spíše k efektivitě než bezpečnosti. Proto se vkládá mezivrstva, která má na starosti zabezpečení jednoduše vytvořených kanálů. Zabezpečovací vrstva se nazývá secure socket layer nebo novější transport layer security. [11]

4.2.2 Secure socket layer a transport layer security

Secure socket layer(SSL) je protokol zajišťující bezpečnost v Internetu. Je to vložená mezivrstva mezi transportní a aplikační vrstvou. Poskytuje zabezpečení komunikace autentizací a šifrováním. Prakticky jej využívá každá webová stránka, která před vlastním názvem používá předponu https. Nejčastěji nachází využití v šifrování e-mailové komunikace. Transport layer security je nadstavba transportní vrstvy a představuje novější verzi protokolu SSL. Průběh ustanovení komunikace mezi serverem a klientem pomocí SSL je podobný jako u Handshake protokolu.4.2.3 [14]

4.2.3 Handshake protokol

Protokol potřesení rukou se skládá z pěti kroků. Prvním krokem protokolu je kontakt mezi klientem a serverem zvaný ClientHallo. Během trvání tohoto kroku probíhá první pokus o kontakt ze strany klienta k serveru. Odesílá se startovací paket dat. Následuje krok s pořadovým číslem dva, ve kterém server odpovídá na pozdrav svým ServerHallo. Ve třetím kroku klient žádá server o jeho digitální certifikát. Ten server odešle a naváže s klientem komunikaci.Při výměně certifikátu si systémy vymění své veřejné klíče a umožňují navázání bezpečné komunikace. Veřejné klíče, čísla p a q, a výsledek matematické operace, osobní klíč na straně A, jsou poslední data, která projdou komunikačním kanálem v nezměněné podobě. V dnešní době se používá kryptografie s infrastrukturou veřejného klíče (PKI). Po úspěšně provedeném potřesení rukou už probíhá veškerá komunikace mezi klientem a serverem v zašifrované podobě.[15]

4.3 Kryptografické systémy

Už od prvních strategických bojů byla potřeba utajování plánů před nepřítelem. Za tímto cílem se vyvinul celý vědní obor kryptografie. Na druhé straně, již od prvních nesmělých krůčků, stojí lidé bažící po utajovaném tajemství, zvaní kryptoloanalytici. Souboj se vleče jako tajemná nit staletími. Až vynálezem počítače dostal vědecký rozměr. V dnešní době je možné získat jakákoliv data bez omezení vzdálenosti či limitace dostupností. Svět se spojil promocí webové sítě. A je palčivou otázkou dnešní doby, která data se hodí pro kteréhokoliv uživatele a která ne. Pro zabezpečení dat na internetové síti slouží zabezpečené komunikační kanály ve vsunuté vrstvě protokolu TCP/IP.[9]
Pro zabezpečení dat je nezbytné pozměnit jejich podobu před zasíláním. K tomuto účelu používáme kryptografické systémy. Obecně pro zprávu Z,klíč K,kódující funkce f_E a zašifrovaný text C,platí:

$$C = f_E(K, Z) \tag{4.1}$$

Podle typu funkce f_E dělíme systémy na kryptografické stroje se symetrickou a asymetrickou šifrou. Rozdíl mezi nimi spočívá ve funkci použité pro vytvoření klíče. Systémy s asymetrickou šifrou mají klíč párový(osobní a veřejný), využívají infrastruktury veřejného klíče a jejich funkce f_E není řešitelná v reálném čase. U symetrických strojů je tomu naopak a f_E je řešitelná v reálném čase.

4.3.1 Kryptografie se symetrickou šifrou

Je matematicky podstatně jednodušší. Funkce f_E je řešitelná v reálném čase, za pomoci jediného klíče. I když se jedná o mocný kryptografický nástroj, jehož bezpečnost lze zajistit mnoha vhodnými matematickými funkcemi, nastává zde problém s distribucí klíče. Pokud by totiž nezabezpečeným kanálem procházel zašifrovaný text i klíč, dala by se konverzace odposlechnout a snadno dešifrovat. Některé přístroje obchází problém s distribucí symetrického klíče vytvořením dalšího komunikačního kanálu.
Symetrické šifrovací funkce se dělí na blokové a proudové. Jako příklad blokové funkce můžeme použít první, zveřejněnou v roce 1977, DES (Data Encryption Standard). Tato šifra má blok o velikosti 64 bitů,z nichž pouze 56 je funkčních. Bere si z původního textu část o velikosti klíče, s tou provádí řadu matematických úprav, podle Feistelovy sítě, a teprve po dokončení úprav v jednom bloku, začne zpracovávat blok další.
Příkladem proudové šifry je RC4, stvořené Riverstem (jedním za autorů RSA). Tato šifra generuje pseudonáhodný proud bajtů a ten dále náhodně(za pomoci operace XOR) pospojuje s čistým textem. Využívá klíč o velikosti 256 bajtů (2048 bitů).

4.3.2 Systémy s asymetrickou šifrou

Funkce f_E v systémech s asymetrickou šifrou představuje časově velmi náročné řešení. Šifra se dělí na dva neoddělitelné klíče, veřejný a osobní.Typickým představitelem asymetrické šifry je RSA. Jako první se používá jak k šifrování tak i k podepisování. Funguje na podobném principu jako Diffieho-Hellmanova výměna klíčů, jen je rozšířená o Eulerovu funkci a výpočty pomocí Euklidova algoritmu.

4.3.3 Diffieho-Hellmanova výměna klíčů

Američtí vědci Whitfiel Diffie a Martin Hellmann vymysleli bezpečný způsob distribuce klíčů. Jeho základ je použití dosavadní nemožnosti faktorizace velkých prvočísel. Při použití Diffieho-Hellmanovy výměny klíčů si klient kdykoliv může zjistit veřejný klíč serveru, s veřejným klíčem provést několik základních matematických operaci a stvořit tak soukromý klíč, ke kterému nebude mít nikdo jiný přístup. Vytvoří tak prostor pro neodposlechnutelnou a nepozměnitelnou komunikaci. Metoda PKI (privat key infrastructure) vyřešila problém s distribucí klíčů svérázným způsobem. Část klíče odtajnila. Na každé straně komunikace pak probíhá sada matematických operací s veřejným klíčem. Veřejný klíč, v případě Diffie-Hellmannovy metody, představuje dostatečně velké prvočíslo. Neschopností faktorizace takto velkých čísel je zaručena tvorba soukromého klíče. Soukromý klíč se již přímo využívá pro výpočet klíče k šifrování zprávy. Hellman a Diffie přišli na geniální nápad s počítáním pomocí matematické funkce modulo. Celočíselný zbytek(modulo) vykazuje stejný výsledek pro velké množství kombinací čísel dělence a dělitele. Po dělení dostatečně vysokého prvočísla, umocněného na "tajnou esenci" a, plynule přechází v tajnou zprávu A. Obě strany vystavují své veřejné klíče p a q. Klient si odešle své p a q serveru, doplní si výpočet svým tajným a, přesně podle vztahu4.2 1 pro výpočet tajného A:

$$A = q^a mod(p) \tag{4.2}$$

server si počíná obdobně se získanými veřejnými klíči p a q, pouze dosadí vlastní tajnou část b

$$B = q^b mod(p) \tag{4.3}$$

Pro ustanovení klíče stačí provést výměnu výsledků A a B a vypočítat

$$K = B^a mod(q) \tag{4.4}$$

nebo

$$K = A^b mod(p) \tag{4.5}$$

Hodnota K představuje vlastní šifrovací klíč použitelný na obou stranách.
Příklad pro předsravu provádění výpočtu:
1. Zvolím P:

P=67841

2. Zvolím Q:

Q=57179

Musí platit,že p i q jsou prvočísla a současně i q<p. Klient má své tajné číslo a:

a=541

Dosazením do rovnice 4.2 pro výpočet tajného A získám u klientova stroje na

A=55940

Již paket dat obsahujících A je pro případného návštěvníka kanálu s pokusem o odposlech zcela nic nevypovídající číslovka. Server si volí vlastní libovolné číslo b.

b=645

A dopočítá

B=34805.

Za pomocí čísel P,Q a A doplní svůj algoritmus výpočtem rovnice 4.4 pro výpočet klíče.

K=6964

Což v binární soustavě představuje 1101100110100, tedy 13 bitový originální klíč. Stejný výsledek dostane i protistrana při stejném postupu. Příklad je pouze ukázkový, protože v dnešní době se za bezpečné považují klíče o velikosti 1024 nebo 2048 bitů.[16] Jelikož by se jednalo o časově velmi náročnou operaci, je reálně prováděna s kratším kódem blokovým způsobem v rámci grupy 1024 nebo 2048 bitů. [15][12]

5 VLASTNÍ APLIKACE

Pro ověření teorie uvedené v kapitole 4 je sestavena aplikace v jazyce Java. Aplikace je model kryptografického systému na bázi klient-server.První dialog vyzve klienta, aby zadal IP adresu serveru, na který se chce přihlásit. Server očekává na předem stanoveném portu přihlášení klienta. Na serveru je po dobu několika sekund otevřen port XY a server pomocí smyčky (klíčové slovo **try**) testuje přihlášení klienta:

```
public void run()
  {
     while(true)
     {
        try
        {
           System.out.println("Naslouchám na portu " +
           serverSocket.getLocalPort() + ".");

     Socket server = serverSocket.accept();
           System.out.println("Prave jsem se pripojil k "
                 + server.getRemoteSocketAddress());
```

Pokud se povede spojení v požadovaném časovém limitu, který ošetřuje metoda **.setSoTimeout** je vše připraveno pro výměnu dat. Časový údaj limitu je nastavený primárně v milisekundách.

```
{
      serverSocket = new ServerSocket(port);
      serverSocket.setSoTimeout(100000);
   .
   .
   .
   .

    }catch(SocketTimeoutException s)
         {  System.out.println("Cas vyprsel!");
            break;
   }
```

Po přihlášení klienta a ověření jeho údajů (metoda **client.getRemoteSocketAddress** z pozdravu klienta směrem k serveru "client hallo") je zahájena komunikace. Proběhne výměna údajů pro výpočet klíče podle Diffieho a Hellmana. Klient odesílá své

veřejné klíše P a Q serveru za pomoci metody(**out.write()**). K odeslaným datům přidává i svoje osobní tajemství OA.

```
 //vystupni proud dat
        OutputStream outToServer = client.getOutputStream();
        DataOutputStream out =new DataOutputStream(outToServer);
      //client hallo
        out.writeUTF("Ahoj,zdraví:"
                + client.getLocalSocketAddress());
       //odeslani dat pro vypocet klice
        out.writeDouble(p);
        out.writeDouble(q);
        out.writeDouble(OA);
```

Pomocí algoritmu (z rovnice4.4) asymetrické šifry vznikne klíč K.

```
//vypocet klice
double k2=Math.pow(OA,b);
double KB=k2 % p;
            System.out.println("Klic pro tento prenos je " + KB);
```

Klíč je bezprostředně po svém ustanovení převeden do binární podoby a upravuje se do velikosti jednoho bloku.

```
//blokova funkce kodu,blok o 8 znacich
             int Klic=(int)KB;
             String kod2=Integer.toBinaryString(Klic);
             for (int i=0; i<8; i++)
             {if (kod2.length() < 8 )
                 kod2="0"+kod2; }
               kod2 = kod2+" ";
               for (int i=0;i<=(C.length())/9;i++)
              {kod2=kod2+kod2;}
               String kod3;
               kod3 = kod2.substring(0, kod2.length());
```

Data z klíče a zprávy pro zašifrování se převedou do datového typu byte. Byte je vstupní datový typ pro metodu xor, která provádí na vstupu matematickou operaci XOR. Xor je matematicky nonekvivalence, jejíž pravdivostní tabulku uvádí tabuka5 1:.

X	Y	X xor Y
1	1	0
1	0	1
0	1	1
0	0	0

Zvolená metoda xor je speciálně upravena pro provedení operace xor na řetezcích String. Provádí binární operaci xor zvlášť na každé pozici dodaných řetězců. Metoda pro výpočet funkce XOR probíhá na obou stranách, tedy u klienta i serveru. Na straně klienta plní šifrovací funkci a na straně serveru dešifrovací. Funkce je plně komutativni, proto na pořadí vstupů nezáleží.

```
//funkce pro operaci xor na dvou proměnných ve formatu string
   public static byte[] xor(final byte[] input, final byte[] secret) {
    final byte[] output = new byte[input.length];
//chyba pri nezadani vstupu
    if (secret.length == 0) {
        throw new IllegalArgumentException("Prazdny klic!!!");}
//operace xor na kazde pozici
    int spoz = 0;
    for (int poz = 0; poz < input.length; ++poz) {
        output[poz] = (byte) (input[poz] ^ secret[spoz]);
        ++spoz;
        if (spoz >= secret.length) {
            spoz = 0;
        }
    }
    return output;
}
}
```

Výsledkem této operace je na straně klienta zašifrovaná zpráva (tj. funkce XOR mezi příslušným bitem klíče a zprávy) a na straně serveru zpráva odšifrovaná, původní. Jelikož z principu funkce XOR probíhá šifrování a dešifrování na úrovni jednotlivých bitů, tj. vztah 1 bit klíče a 1 bit zprávy, nemůže dojít ke ztrátě významu dané zprávy, protože práce s jednotlivými bity není závislá na použitém kódování zprávy operačního systému. Ke zkreslení může dojít pouze na straně operačního systému serveru nebo klienta, kdy, v případě rozdílné platformy (Windows - CP1250 vs. Linus ISO-8859-2 nebo UTF-8) může být zpráva uložena jinou ASCII tabulkou.
V navržené aplikaci docházelo k problému s ustanovením stejného klíče, protože

zvolený datový typ double má tendenci k zaokrouhlování velkých čísel. Toto lze vyřešit novým během programu po vygenerování chybové hlášky.

6 OVĚŘENÍ ŘEŠENÍ

Kapitola ukazuje výsledky z jednoho běhu programu.

6.1 Server

Naslouchám na portu 2000.
Prave jsem se pripojil k /147.229.187.34:50546 | (Povedlo se spojení)
Ahoj,zdraví:/147.229.187.34:50546 |(Data obdržená z ClientHallo)
P 6113.0
Q 373.0
OA 5144.0
(Data obdržená k ustanovení klíče)
OB 373.0|(Osobní B)
Klic pro tento prenos je 5144.0 |(Vytvořený klíč)
Binary 10100000 11000101 00000110 00101000 00110001
01000001 10001010 00001100 01010000
01100010 10000011 00010100 00011000 10100000
11000101 00000110 |(Klíč po blokové úpravě)
SAL 01010100 01100101 01110011 01110100 01101111 01110110
01100001 01100011 11101101
00100000 01111010 01110000 01110010 11100001
01110110 01100001|(Odšifrovaná zpráva)
T
e
s
t
o
v
a
c
í| (převod z binární reprezentace na písmena dle UTF-8)
z
p
r
á
v
a

6.2 Klient

Na straně klienta je využita následující tabulka6.2 pro překlad zprávy.

Písmeno	Binární reprezentace	Kód	Výsledek
T	01010100	10100000	11110100
e	01100101	11000101	10100000
s	01110011	00000110	01110101
t	01110100	00101000	01011100
o	01101111	00110001	01011110
v	01110110	01000001	00110110
a	01100001	10001010	11101011
c	01100001	00001100	01101101
í	11101101	01010000	10001101

Aplikace Klient dále vypíše:
Zadejte adresu IP a potvrďte tlačítkem enter:|
147.229.196.130|(Vložení adresy serveru)
Testovací zpráva
P = 6113.0
Q = 373.0
OA =5144.0|(Vygenerovaná data)
Pripojuji se na 147.229.196.130 na portu 2000|(Zahájení spojení)
Prave jsem se pripojil na /147.229.196.130:2000// |(Spojení se serverem proběhlu úspěšně)
OB 373.0 |(Osobní výpočet B)
Klic je 5144.0|(Ustanovení klíče)
Server povida: Dekuji za pripojeni na /147.229.196.130:2000
Do videni!|(Ukončení spojení)
Běh toto programu úspěšně přenesl testovací zprávu, ustanovil klíč K=5144,podle kterého ji zašifroval i odšifroval. Vše proběhlo v rámci několika sekund.

7 ZÁVĚR

Práce se zabývá tématem datových a komunikačních standardů ve zdravotnictví. V první kapitole jsem shrnula dostupné poznatky o zdravotnických datových a komunikačních standardech. Popisuji vývoj českého datového standardu od DS 1.0 k dnešnímu DS 4, kde největší změnou je struktura XML. Nutné bylo i srovnání s celosvětově nejvíce rozšířeným standardem Health level seven (HL7). Dnešní medicína již využívá různých zobrazovacích zařízení a proto je opět nutné standardizovat jejich výstupy. Příkladem digitálního obrazového standardu je DICOM.
Druhá kapitola se zaměřuje na reprezentaci medicínského záznamu, kde kódy dosazené pomocí systému MKN nebo MKF snižují časové nároky na výpis každé zprávy pro pacienta. Jelikož se jedná o mezinárodní kódy, usnadňují i práci s ošetřením pacienta v jiné zemi bez nutnosti překladu anamnézy. Jsou použitelné i pro administrativní činnosti a posudky zcela mimo oblast zdravotnictví, ale zasahující až do státní správy sociálního zabezpečení. Příkladem takového využití by mohla administrativa spojená s podáním žádosti o invalidní důchod.
Ve třetí kapitole popisuji národní informační systém a jeho podčásti, jednotlivé registry. Vše shrnuje organizační diagram na obrázku 3.
Čtvrtá kapitola patří zabezpečení datové komunikace. Je zde popsán internetový protokol TCP/IP v porovnání s ideálním modelem pro přenos pomocí OSI/ISO. V rámci jednotlivých skupin protokolů bezpečnosti v TCP/IP jsem si dovolila vyzvednout dva nejdůležitější a to SSL a jeho následovníka TLS. Dále je v kapitole zmíněno šifrování pomocí asymetrického i symetrického klíče. Speciální část je věnována výměně klíčů podle Whitfielda Diffie a Martina Hellmanna. Tato průlomová metoda ze 70.let minulého století je originální svým řešením problému distribuce klíčů při šifrování dat. Pomocí velkých prvočísel a sady matematických operací je na obou stranách komunikace ustanoven klíč, který ale ani jedinkrát neprojde v žádné podobě komunikačním kanálem.
Cílem práce bylo ověřit získané poznatky na kryptografickém modelu pro přenos datových zpráv. Model byl navržen a dále i realizován v programovacím jazyce Java. V modelu se povedlo ověřit, že přenos dat mezi serverem a klientem je možný. Výměna klíčů podle Diffie a Hellmana podle teoretických poznatku získaných v kapitole 4.3.3 je plně funkční záležitost. Šifrování nezabere mnoho času a předejde velkým nepříjemnostem.

LITERATURA

[1] *Mezinárodní klasifikace funkčních schopností, disability a zdraví: MKF.* 1. české vyd. Překlad Jan Pfeiffer, Olga Švestková. Praha: Grada, 2008, 280 s. ISBN 978-80-247-1587-2.

[2] MÜNZ, J. *Informační technologie ve zdravotnictví: informační systémy.* 1. vyd. V Praze: České vysoké učení technické, 2011, 304 s. ISBN 978-80-01-04720-0.

[3] *Ústav zdravotnických informací a statistiky ČR: Webová prezentace.* ÚZIS ČR 2010-2012. ÚZIS ČR: Registry a informační systém NZIS [online]. Praha, 2012 [cit. 2012-12-10]. Dostupné z URL: <`http://www.uzis.cz/`>.

[4] HUSEBY, S. H. *Zranitelný kód.* Vyd. 1. Brno: Computer Press, 2006, 207 s. ISBN 80-251-1180-6.

[5] ÚSTAV ZDRAVOTNICKÝCH INFORMACÍ A STATISTIKY ČESKÉ REPUBLIKY. *Mezinárodní klasifikace nemocí: Instrukční příručka* [online]. 10.revize. Praha, 2009, 1.1.2009 [cit. 2012-12-13]. Dostupné z:<`http://www.uzis.cz/cz/mkn/index.html`>

[6] LUKEŠ, D. *HTTPS - bezpečnost jen pro vyvolené?.* In: Lupa.cz: Server o českém Internetu [online]. 22.2.2001. 2001 [cit. 2013-03-20]. Dostupné z:<`http://www.lupa.cz/clanky/https-bezpecnost-jen-pro-vyvolene`>

[7] HANÁČEK, P., STAUDEK,J. *Bezpečnost informačních systémů: metodická příručka zabezpečování produktů a systémů budovaných na bázi informačních technologií.* Praha: Úřad pro státní informační systém, 2000, 127 s. ISBN 80-238-5400-3.

[8] DOSTÁLEK, L.,VOHNOUTOVÁ M.,KNOTEK.M. *Velký průvodce infrastrukturou PKI a technologií elektronického podpisu.* 2. aktualizované vyd. Brno: Computer Press, 2009, 542 s. ISBN 978-80-251-2619-6.

[9] SINGH, S. *Kniha kódů a šifer: tajná komunikace od starého Egypta po kvantovou kryptografii.* 2. vyd. v čes. jaz. Překlad Petr Koubský, Dita Eckhardtová. Praha: Dokořán, 2009, 382 s. Aliter, sv. 9. ISBN 978-802-5701-447.

[10] HELLMAN,et al. STANFORD UNIVERSITY. *Cryptographic aparatus and method* [patent]. United States of America. United States Patent, 4200770. Uděleno 29.4.1980. Dostupné z:<`http://www.google.com/patents/US4200770`>

[11] PUŽMANOVÁ, R. *TCP/IP v kostce.* 2. upravené a rozšířené vyd. České Budějovice: Kopp, 2009, 619 s. ISBN 978-80-7232-388-3.

[12] STALLINGS, W. *Cryptography and network security*. Vyd. 1. New Jersey: Prentice-Hall, 1999, 569 s. ISBN 01-386-9017-0.

[13] PETERKA, J. *Současné trendy ve světě protokolů TCP/IP*. Earchiv.cz: archiv článků a přednášek Jiřího Peterky [online]. 2011 [cit. 2013-05-22]. Dostupné z:<http://www.earchiv.cz/a96/a636k150.php3>

[14] DIERKS, T. RESCORLA, E. *RFC 5246. The Transport Layer Security (TLS) Protocol*. RTFM, Inc., 2008. Dostupné z: <http://tools.ietf.org/html/rfc5246>

[15] RFC 2631. *Diffie-Hellman Key Agreement Method*. RTFM,Inc, 1999. Dostupné z:<http://www.ietf.org/rfc/rfc2631.txt>

[16] ECRYPT II Yearly Report on Algorithms and Keysizes (2011-2012) [online]. 1.revize. Katholieke Universiteit Leuven (KUL), 2012, 30.9.2012 [cit. 2013-05-28]. Dostupné z: <http://www.ecrypt.eu.org/documents/D.SPA.20.pdf>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

ANSI	American National Standards Institute
ČSSZ	Česká správa sociálního zabezpečení
ČSÚ	Český statistický úřad
DaSta	Datový standard ministerstva zdravotnictví ČR
DICOM	Digital Image and Communications In Medicine protocol suite
DS	Datový standard
EODS	European Occupational Diseases Statistics
HL7	Health Level Seven
IACR	Mezinárodní sdružení onkologických registrů
IP	Internet Protocol
MKF	Mezinkárodní klasifikace funkčních schopností, disability a zdraví
MKN	Mezinárodní klasifikace nemocí
MZ ČR	Ministerstvo zdravotnictví České republiky
NČLP	Národní číselník laboratorních pomůcek
NIS	Nemocniční informační systémy
NOR	Národní onkologický registr
NRHOSP	Národní registr hospitalizovaných
NRKN	Národní registr kloubních náhrad
NRKOI	Národní registr kardiologických operací a intervencí
NRLUD	Národní registr léčby uživatelů drog
NRNP	Národní registr nemocí z povolání
NROVDK	Národní registr osob vyloučených z dárcovství krve
NRPTV	Národní registr pitev a vyšetření prováděných na toxikologickém oddělení

NRRZ	Národní registr reprodukčního zdraví
NRU	Národní registr úrazů
RZS	Rychlá záchranná služba
SQL	Structured Query Language
SSH	Save Shell
TCP	Transission Control Protocol
USA	United States of America
ÚZIS	Ústav zdravotnických informací a statistiky
WHO	World Health Organization
XML	Extensible Markup Language

SEZNAM PŘÍLOH

A PŘÍLOHY

A.1 Kód kryptografického systému

A.1.1 Klient

```
      import java.net.*;
import java.util.Random;
import java.io.*;
import java.util.Scanner;
public class Klient
{
   public static void main(String [] args)

 //vstup pro zadani adresy IP serveru
   { Scanner ip=new Scanner (System.in);
   String serverName;
   System.out.println
   ("Zadejte adresu IP a potvrďte tlačítkem enter:");
   serverName = ip.nextLine();

 //nacteni zpravy
   StringBuilder k=nacitani("ahoj.txt");
   String w=k.toString();

 //volba P
      double p;
      p=volba();

 //overeni prvociselnosti
      boolean m=false;
      m=Prvocisla(p);
      while (m==true)
  {p=volba();
  m=Prvocisla(p);}
    System.out.println("P = "+p);

 //volba q
  boolean o=false;
```

```
 double q;
 do { q = volba();
} while(p<q);
o=Prvocisla(q);
while (o==true || p<q)
{q=volba();
o=Prvocisla(q);}
   System.out.println("Q = "+q);

//esencialni a
 int a;
 Random f = new Random();
 a = f.nextInt(20);

//vypocet osobniho tajneho OA
 if(a<p){
 double k1= Math.pow(q,a);
 double OA =k1%p;
 System.out.println("OA ="+  OA);

//vyber portu
 int port = Integer.parseInt("2000",10);

//cyklus pro pripojovani
 try
 {  System.out.println("Pripojuji se na " + serverName
                            + " na portu " + port);

//nadefinovani typu spojeni
      Socket client = new Socket(serverName, port);

//vystupni data
       System.out.println("Prave jsem se pripojil na "
                     + client.getRemoteSocketAddress());
       OutputStream outToServer = client.getOutputStream();
       DataOutputStream out =new DataOutputStream(outToServer);

//client hallo
       out.writeUTF("Ahoj,zdraví:"
```

```
                + client.getLocalSocketAddress());

 //odeslani dat pro vypocet klice
        out.writeDouble(p);
        out.writeDouble(q);
        out.writeDouble(OA);

//prichozi data
        InputStream inFromServer = client.getInputStream();
        DataInputStream in = new DataInputStream(inFromServer);
         double OB;
         OB=in.readDouble();
         System.out.println("OB "+OB);
         double k3= Math.pow(OB,a);
         double KA=k3%p;
         System.out.println("Klic je "+KA);
         int Klic=(int)KA;

 //prevod klice na binarni kod
         String bin=Integer.toBinaryString(Klic);
         String bin1=bin;

 //sestaveni blokove funkce klice
         for(int i=0; i < 8; i++)
          {if ( bin.length() < 8 )
          bin = "0" + bin;}
          bin=bin+" ";
         for(int i=0; i<=(w.length()-8)/9; i++)
         { bin=bin+bin;}
         bin=bin+bin1;

//priprava dat pro sifrovani
          byte[] sifra=w.getBytes();
          byte[] klic=bin.getBytes();

//sifrovani pomoci xor
          byte[] z=xor(sifra,klic);

//vysledek preveden do retezce string
```

```
     String zpr = new String(z);

//zakodovana zprava
     System.out.println(zpr);
     System.out.println("Server povida: " + in.readUTF());

//odeslani dalsiho paketu dat
    DataOutputStream odch =new DataOutputStream(outToServer);
    odch.writeUTF(zpr);

 //ukonceni klienta
     client.close();
      }catch(IOException e)
      {
         e.printStackTrace();
      } }
  ip.close();}

//metoda pro volbu cisla
    public static double volba(){
   double k;
Random r = new Random();
k =r.nextInt(100);
   return k;}

//metoda overeni prvociselnosti
    public static boolean Prvocisla(double k){
boolean neni = false;
int i=2;

//testuje delitelnost do velikosti poloviny daného cisla k
if (k > 2) {
do { neni = ((k % i) == 0);
i++;
} while ((!neni) && (i < (k / 2)));
    }
         return neni;}

//metoda pro nacitani zpravy
```

```
public static StringBuilder nacitani(String nazev)
{ String sFileName = nazev;
try { String s;
 BufferedReader brFile = new BufferedReader(new FileReader(sFileName));
    while ((s = brFile.readLine()) != null)
     {
//vypis zpravy
      System.out.println(s);

//uprava do binarni podoby
     byte[] bytes =s.getBytes();
        StringBuilder binary = new StringBuilder();
        for (byte k : bytes)
        {  int val = k;
           for (int i = 0; i < 8; i++)
           { binary.append((val & 128) == 0 ? 0 : 1);
              val <<= 1;}
           binary.append(' ')}
       return binary;}

//uzavreni souboru
    brFile.close();}

//vyjimky
         catch (IOException e) {
System.out.println("I/O error in: " +
  sFileName + "\n" + e.getMessage());
}
       return null;}

//funkce pro operaci xor na dvou retezcich
    public static byte[] xor(final byte[] input, final byte[] secret)
         {final byte[] output = new byte[input.length];
         if (secret.length == 0) {
         throw new IllegalArgumentException("Prazdny klic!!!");}
         int spos = 0;
     for (int pos = 0; pos < input.length; ++pos) {
         output[pos] = (byte) (input[pos] ^ secret[spos]);
         ++spos;
```

```
if (spos >= secret.length) {
      spos = 0;}}
    return output;
     }
  }
```

A.1.2 Server

```
import java.net.*;
import java.util.Random;
import java.io.*;

public class server2 extends Thread
{
   private ServerSocket serverSocket;

   public server2(int port) throws IOException
   {
      serverSocket = new ServerSocket(port);
      serverSocket.setSoTimeout(100000);
   }

   public void run()
   {

 //testovani spojeni
      while(true)
      {
         try
         {
            System.out.println("Nasloucham na portu " +
            serverSocket.getLocalPort() + ".");

      Socket server = serverSocket.accept();
      System.out.println("Prave jsem se pripojil k "
                  + server.getRemoteSocketAddress());
      DataInputStream in =new DataInputStream(server.getInputStream());
      System.out.println(in.readUTF());

 //ziskani poslanych dat
            double p;
            p=in.readDouble();
            System.out.println("P "+p);

            double q;
```

```
            q=in.readDouble();
            System.out.println("Q "+q);

            Double OA=in.readDouble();
            System.out.println("OA " +OA);

 //volba b
            int b;
    Random l = new Random();
b = l.nextInt(16);
System.out.println("B "+b);

//vypocet tajneho OB
if (b<p){
double k1=Math.pow(q,b);
double OB=k1%p;
System.out.println("OB "+OB);

//vypocet klice
double k2=Math.pow(OA,b);
double KB=k2 % p;
            System.out.println("Klic pro tento prenos je " + KB);
            int Klic=(int)KB;
            String kod2=Integer.toBinaryString(Klic);

//odeslani dat ke klientovi
       DataOutputStream out =new DataOutputStream(server.getOutputStream());
              out.writeDouble(OB);
              out.writeUTF("Dekuji za pripojeni na "
              + server.getLocalSocketAddress() + "\n Do videni!");

//prichozi data pro zakodovanou zpravu
      DataInputStream in2 =new DataInputStream(server.getInputStream());
             String C=in2.readUTF();

//blokova funkce kodu,blok o 8 znacich
             for (int i=0; i<8; i++)
             {if (kod2.length() < 8 )
                 kod2="0"+kod2; }
```

```
                kod2 = kod2+" ";
                for (int i=0;i<=(C.length())/9;i++)
               {kod2=kod2+kod2;}
                String kod3;
                kod3 = kod2.substring(0, kod2.length());

  //uprava dat pro sifrovaci funkco xor
             byte[] sifra=C.getBytes();
             byte[] klic=kod3.getBytes();

  //volani metody xor
             byte[] z=xor(sifra,klic);

  //prevedeni vysledku do binarni podoby podoby
             String sal = new String(z,"UTF-8");

 //zobrazeni binarni zpravy
             System.out.println("SAL "+ sal);

 //odstraneni mezer
             String newstr = sal.replaceAll(" ", "");

 //cyklus pro preklad binarniho cisla na pismeno dle UTF-8
           for (int i=0; i<=newstr.length();i+=8)
            {String novy2 = newstr.substring(i,i+8);
            int charkod = Integer.parseInt (novy2 ,2);
           char pism=(char)charkod;

//samotny prevod
           String zakl = Character.toString(pism);
     // StringBuffer zakl2 = new StringBuffer();

//Zobrazeni vysledku
          System.out.println(zakl);
          }

//uzavreni vlakna
             server.close();}
```

```
//kontrola casu
}catch(SocketTimeoutException s)
       {
            System.out.println("Cas vyprsel!");
            break;

//kontrola vstupu/vystupu
         }catch(IOException e)
         {
            e.printStackTrace();
            break;
         }
      }
      }

 //nacteni portu
   public static void main(String [] args)
   {
      int port = Integer.parseInt("2000");
      try
      {

  //vlakno s volbou portu
         Thread t = new server2(port);
         t.start();
      }catch(IOException e)
      {
         e.printStackTrace();

      }
   }

//funkce pro operaci xor na dvou proměnných ve formatu string
   public static byte[] xor(final byte[] input, final byte[] secret) {
    final byte[] output = new byte[input.length];

//chyba pri nezadani vstupu
    if (secret.length == 0) {
        throw new IllegalArgumentException("Prazdny klic!!!");}
```

```
//operace xor na kazde pozici
    int spoz = 0;
    for (int poz = 0; poz < input.length; ++poz) {
        output[poz] = (byte) (input[poz] ^ secret[spoz]);
        ++spoz;
        if (spoz >= secret.length) {
            spoz = 0;
        }
    }
    return output;
}
}
```