Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Kryptografická ochrana osobných údajov v cloude

Diplomová práca

2020
Bc. Lenka Stúpalová

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

Kryptografická ochrana osobných údajov v cloude

Diplomová práca

Študijný program:	Informatika
Študijný odbor:	Informatika
Školiace pracovisko:	Katedra informatiky
Školiteľ:	RNDr. Michal Rjaško PhD.

Bratislava, 2020
Bc. Lenka Stúpalová

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Bc. Lenka Stúpalová
Študijný program: informatika (Jednoodborové štúdium, magisterský II. st., denná forma)
Študijný odbor: informatika
Typ záverečnej práce: diplomová
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Kryptografická ochrana osobných údajov v cloude
Cryptographic protection of personal information in cloud

Anotácia: Rozšíreniu cloudových riešení typu software as a service do podnikov a štátnej správy častokrát bráni požiadavka na ochranu osobných údajov uložených v cloude.
Cieľom práce je preskúmať a analyzovať existujúce možnosti a navrhnúť vhodné riešenie, ktoré by zabezpečilo kryptografickú ochranu osobných údajov spracovávaných v cloude.

Vedúci: RNDr. Michal Rjaško, PhD.
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Dátum zadania: 11.12.2017

Dátum schválenia: 11.12.2017 prof. RNDr. Rastislav Kráľovič, PhD.
garant študijného programu

.....
študent

.....
vedúci práce

Poďakovanie: Chcela by som poďakovať svojmu školiteľovi, RNDr. Michalovi Rjaškovi, PhD., za jeho pomoc, usmerňovanie, odborné rady a trpezlivosť. Taktiež by som chcela poďakovať oponentovi prvej verzie tejto diplomovej práce, Doc. RNDr. Danielovi Olejárovi, PhD, za jeho záujem o zdokonalenie práce, rady a návrhy na vylepšenie práce.

Abstrakt

Používanie cloudových služieb v posledných rokoch stále narastá. Je to spôsobené početnými výhodami cloudu. Avšak bezpečnosť a dôvernosť dát uložených na takomto verejnom úložisku je neistá. V našej práci sa venujeme práve zabezpečeniu bezpečnosti takýchto dát a osobných údajov v cloude. Popísali sme vlastnosti cloudov. Rozobrali sme zákony SR o ochrane osobných údajov a zistili sme tak povinnosti prevádzkovateľa pri spracovávaní osobných údajov. Ďalej sme popísali metódy, ktoré už existujú a ktoré slúžia práve na zabezpečenie ochrany osobných údajov v cloude. Týmito boli homomorfické šifrovanie, CryptDB, C-SDA, GhostDB. Spomenuli sme aj knižnice využívajúce homomorfické šifrovanie, ktorou je napríklad knižnica Helib. Spomenuté systémy často znižovali silu šifrovania dát pre väčšiu efektívnosť vykonávania dotazov na nich. V poslednej časti rozoberáme naše návrhy riešenia. V návrhoch v prvom rade kladieme dôraz na bezpečnosť údajov a súlad spracovania osobných údajov so zákonmi a normami Slovenskej republiky. Efektívnosť je pre nás druhoradá a v práci sa jej veľmi nevenujeme. V návrhoch je použité najmä šifrovanie AES-256.

Kľúčové slová: cloud computing, Zákon o ochrane osobných údajov, šifrovanie dát v cloude, homomorfické šifrovanie, lokálne šifrovanie, bezpečnosť osobných údajov

Abstract

We can see a growth of the usage of cloud systems in recent years. This is due to cloud's numerous advantages. However, the security of data stored in a public repository is questionable. In our work we are focusing on ensuring this security of data and personal informations in cloud. In the first part we described the properties of a cloud. Then we analyzed laws of Slovak Republic on protection of personal data. This laws are implementing regulations from European Union's GDPR. We found out, which duties people processing personal data have. In next part we described the existing methods for ensuring protection of personal data. These methods were: Homomorphic encryption, CryptDB, C-SDA and GhostDB. We also mentioned libraries implementing homomorphic encryption such as Helib. These systems often drop on security for more efficient query execution. In the last part we discuss our proposals. In proposals, we are enforcing data security and compliance with laws of the Slovak Republic over efficiency. The proposals mainly use AES-256.

Keywords: cloud computing, data protection laws in Slovakia, encryption in cloud, homomorphic encryption, local encryption, safety of personal data

Obsah

Zoznam obrázkov

Zoznam tabuliek

Úvod

Cloud computing je model, ktorý dovoľuje ľuďom vypožičať si externe hardvér a softvér cez internet a používať ho tak na diaľku či už ako svoje rozšírené úložisko alebo ako výpočtovú silu. V posledných rokoch sa stáva táto služba veľmi obľúbenou a využívanou ako jednotlivcami, tak aj mnohými firmami či vládou. Tento trend je spôsobený mnohými a významnými výhodami cloudov ako je napr. celosvetová dostupnosť (dát/softvéru), odbremenenie používateľa od fyzického spravovania zariadení, flexibilita cloudu a jednoduché a rýchle prispôsobenie sa našim potrebám a mnohé iné. Problém ale nastáva, ak nechceme na cloud umiestniť verejnú aplikáciu, ale chceme si tu len uložiť nejaké súkromné dáta či firemné údaje, prípadne vláda chce mať databázu občanov štátu, neplatičov daní s ich osobnými údajmi. Typickým a dobrým príkladom citlivých informácií spravovanými v cloud computing prostredí sú zdravotné záznamy. Je celkom zjavné, že väčšina ľudí chce, aby ich zdravotné problémy boli chránené. Sú ale takéto údaje v cloude zabezpečené? Bezpečnosť citlivých údajov v cloude je otázna, pretože tieto dáta sú uchovávané a spracovávané pomocou aplikácií tretej strany. Tieto dáta sú taktiež uložené na zdieľanom hardvéri v cloude, spolu s dátami iných zákazníkov a firiem, čo sa môže zdať útočníkom viac atraktívne a prilákať ich pozornosť. Takto by útočníkovi stačilo hacknúť jednu službu a získal by dáta od veľkého množstva firiem využívajúcich ten istý cloud.

Ako zabezpečiť údaje pri ich uchovávaní na cloude? Určite by sme ich mali minimálne prenášať šifrovane medzi používateľom a cloudom, čo je v súčasnosti štandardom - najčastejšie sa ku cloudu pristupuje cez zabezpečený protokol https. Údaje ale môžu byť v cloude uložené v otvorenom texte a útočník alebo osoba s prístupom k databáze sa k nim vie dostať. Veľa cloudových systémov ani dáta na cloude nešifruje, pretože to uberá na efektívnosti a rýchlosti prístupu a komunikácie používateľov s cloudom a väčšine ľudí záleži v prvom rade práve na tom a bezpečnosť je až druhoradá. Môžeme ale pridať autentifikáciu používateľov pri požiadavke o prístupe ku cloudu a dáta zašifrovať - používateľ sa musí preukázať heslom, ktoré je uložené u poskytovateľa cloudu a tým získa prístup k svojim odšifrovaným údajom. Toto ale tiež nie je úplne bezpečné, pretože sa musíme v tomto prípade spoliehať na poskytovateľa cloud systému a jeho zamestnancov, možno administrátorov, správcov a podobne, ktorí môžu stále pristupovať k našim údajom. Ak takto dáta šifrujeme, musíme vedieť riešiť aj bežné straty a

zabudntia hesiel. Taktiež sa môže k dátam dostať ľahšie nejaký hacker, ktorý využije určitú zraniteľnosť prípadne chybu spôsobenú programátorom, ktorá mu prezradí dáta uložené na cloude. Môžme teda skúsiť zašifrovať si dáta sami v počítači a ukladať ich na cloud takto zašifrované. Jediný kľúč máme my a nik iný ho nepozná. My jediní vieme k dátam pristupovať. Toto je skutočne bezpečné riešenie ak nedôverujeme poskytovateľom týchto služieb ani softvéru cloudu. Je ale toto riešenie praktické?

Nevýhodou takéhoto obyčajného zašifrovania dát je, že vždy keď chceme pristúpiť k dátam (napr. niečo v nich vyhľadať) alebo ich modifikovať (napr. pre databázu ľudí - zmena bydliska, mena, poisťovne alebo len oprava preklepu...), tak musíme stiahnuť celý súbor (resp. celý obsah databázy, ak nevieme, kde sa nachádza hľadaný výraz), odšifrovať celý súbor, vykonať požadované akcie, znovu zašifrovať a poslať celý súbor naspäť na cloud. Príklady zo života - databáza študentov univerzity 1000 ľudí - cca možno 20Mb. Pre databázu občanov SR alebo zdravotné záznamy lekárov - podstatne viac. Internetový obchod ako napr. eBay má aktívne v predaji aktuálne miliardu vecí [58], a to sú terabajty dát.

Pri spracovávaní osobných údajov ale vstupuje do hry aj zákon. Nestačí si len myslieť, že dáta sú v bezpečí. Takéto osobné a citlivé dáta a narábanie s nimi je ošetrené a strážené zákonom. Po príchode nového nariadenia GDPR a jeho integrácii do zákonov platných v Slovenskej Republike v roku 2018 sa sprísnili tieto opatrenia a nastavili sa vysoké pokuty za ich porušenie. Treba preto naplniť požiadavky zákona a dôsledne zabezpečiť súlad spracovávania a uchovávania týchto údajov so zákonom.

Chceli by sme teda niečo, čo zabezpečí dôvernosť dát a v nich obsiahnutých informácií (že sa k nim vieme dostať len my), zároveň aj ich dostupnosť, aby sme k nim vedeli pristupovať a narábať s nimi efektívne a taktiež bude v súlade so zákonmi SR. Pokúsiť sa vytvoriť niečo takéto je jedným z cieľov tejto práce.

V priebehu riešenia práce sa ukázalo, že navrhnúť a implementovať funkčný systém riešenia ochrany osobných údajov v cloude, je úloha prevyšujúca možnosti diplomovej práce. Preto sme celú problematiku detailne popísali, rozobrali existujúce riešenia a naznačili možnosti riešenia vlastnými návrhmi v teoretickej rovine.

V tejto práci teda najskôr v pvej časti čitateľa stručne oboznámime s pár základnými pojmami z kryptológie, priblížime princípy fungovania cloud computingu a jeho využívanie s súčasnosti. V ďalšej časti preberieme zákony o ochrane osobných údajov platné v Slovenskej Republike a zákony týkajúce sa cloudových služieb. (Ďalšie odporúčania, sprievodníky, pokyny a rady týkajúce sa cloud computingu a ochrany dát môžete nájsť v zdrojoch [35], [8] a [33].) Následne si popíšeme doposiaľ používané metódy na ochranu údajov v cloudových službách a pojednáme o ich hlavných nedostatkoch a nevýhodách, pre ktoré nie sú reálne schopné byť zavedené do praxe. V poslednej časti skúsime navrhnúť a rozobrať vlastné riešenia, ktoré budú rozumne vyvažovať bezpečnosť a efektivitu cloud computingu s dôrazom na bezpečnosť.

Kapitola 1

Základné pojmy a úvod do cloud computingu

V tejto kapitole si najskôr povieme stručne základné pojmy z kryptológie. Potom si priblížime pojem cloud computing, čo to je, prečo a načo je táto technológia používaná, akú poskytujú poskytovatelia ochranu údajov v cloudoch a podobne. Taktiež sa pozrieme na nárast využívania cloudových služieb v posledných rokoch.

1.1 Základné pojmy z kryptológie

Objasnenia pojmov sú prebraté z [65].

Šifrovanie - Utajenie správy, prevedenie z jej čitateľnej podoby do nezrozumiteľnej.

Dešifrovanie - Zistenie obsahu utajenej správy, prevedenie správy naspäť do čitateľnej, pôvodnej.

Kľúč - je istá informácia, ktorá určuje *zmenu* správy z čitateľnej na nezrozumiteľnú, resp. naopak pri dešifrovaní.

Symetrické šifrovanie

Symetrické šifrovanie je spôsob šifrovania, ktorý používa taký istý kľúč pre zašifrovanie ako aj následne pre dešifrovanie. Tento spôsob šifrovania je výpočtovo oveľa rýchlejší ako asymetrické šifrovanie aj preto, že (zvyčajne) môže používať kratšie kľúče. V dnešnej dobe stačí kľúč dĺžky 128 bitov. Už takýto dlhý kľúč útočník nie je schopný efektívne vypočítať, pretože potrebuje obvykle vyskúšať všetky možné kľúče a tých je 2^{128}. Samozrejme toto šifrovanie má zmysel len ak predpokladáme, že kľúč na dešifrovanie nepozná nikto cudzí a vlastní ho len príjemca. Preto si musíme dávať pozor na bezpečné

prenesenie a uschovanie kľúča. V súčasnosti sa symetrické šifrovanie využíva na bežné šifrovanie a dešifrovanie diskov, nakoľko nespôsobuje žiadne pocítiteľné spomalenie. Symetrické šifry môžeme ďalej deliť na blokové a prúdové.

Asymetrické šifrovanie

Pri asymetrickom šifrovaní treba na dešifrovanie iný kľúč ako pre zašifrovanie. Existujú teda dva kľúče, jeden verejný a jeden súkromný, ktoré sú zvolené tak, aby po aplikácii oboch na nejakú správu ostala správa v konečnom dôsledku bez zmeny a zároveň nebolo možné efektívne vypočítať z verejného kľúča súkromný. Verejný kľúč sa používa na šifrovanie a súkromný na dešifrovanie.
Táto metóda je výpočtovo náročnejšia, nakoľko je treba oveľa dlhšie kľúče, aby sa súkromný kľúč nedal efektívne vypočítať z verejného. Keby sme chceli rovnako „náročnú zistiteľnosť“ kľúča (teda bezpečnosť) ako pri 128 bitovom kľúči pri symetrickom šifrovaní, potrebovali by sme (pri implementáciách asymetrických šifier založených na faktorizácií integerov, ako napr. RSA šifra) kľúč veľkosti 3072 bitov. Avšak nemusíme zabezpečovať bezpečné prenesenie kľúčov, pretože súkromný kľúč máme len my a znalosť verejného kľúča aj neželanými osobami by nám, pri správnej implementácii, nemala nijako uškodiť.

Blokové šifry

Bloková šifra šifruje správu po blokoch, teda robí operácie nad fixným počtom bitov. Ak je počet bitov väčší ako určený, tak správu rozdelí na časti. (Časti sa môžu, alebo nemusia vzájomne ovplyvňovať.) Ak je naopak správa kratšia ako určený počet bitov, správa sa doplní, nakoľko niektoré blokové algoritmy nevedia robiť s kratšími reťazcami. Toto vyplnenie sa nazýva aj padding.
Blokové šifrovanie je iteratívna šifra, a teda sa pri ňom používa napr. niekoľkonásobná jednoduchá transformácia.

Hashovacie funkcie

Hashovacie funkcie sú funkcie, ktoré vytvoria zo zadaného vstupu reťazec pevnej dĺžky, tzv. odtlačok, pričom nezáleží na tom či je vstup kratší alebo niekoľkonásobne dlhší ako dĺžka odtlačku. Odtlačok sa nazýva hash a zvyčajne mával dĺžku 128 či 160 bitov. Avšak dĺžka hashu sa postupne navyšuje a v novo navrhnutých súčasných algoritmoch môže byť výstup dlhý aj viac bitov, napr. až 256. Na rozdiel od šifrovania tu prichádza k redukcii informácií, preto nejde pôvodnú správu naspäť zrekonštruovať. Hashovanie je teda len jednosmerné. Výpočet hashu je v niektorých funkciách či implementáciách veľmi rýchly, čo je pre ne dobrá vlastnosť. V iných hashovacích funkciách môže byť výpočet hashu pomalý, čo môže byť tiež dobrá vlastnosť, ktorá takto napr. spomalí

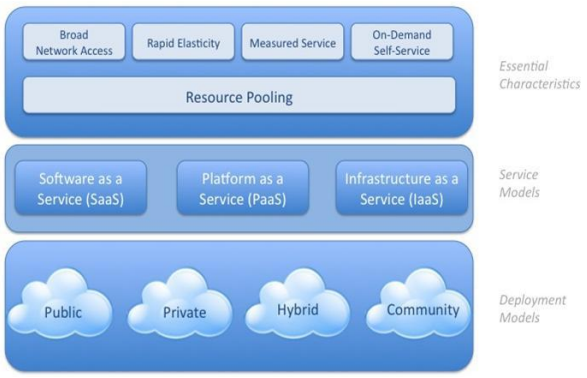
útočníkom brute-force útok (útok hrubou silou, skúšanie všetkých možností). Preto sa hashovacie funkcie hojne používajú napr. na: overenie integrity dát ako kontrolný súčet, detekciu chýb - toto je potrebné nielen na odhalenie zmeny textu útočníkom, ale aj chýb zlyhania techniky či vplyvu prostredia, ktoré nemajú nulovú šancu nastania pri zápise či prenose dát, indexovanie - kde hash reprezentuje akoby vlastnosti dát, bezpečné uloženie hesiel v databázach, rýchle vyhľadávanie.

Tieto funkcie sú deterministické a nepoužívajú žiadne kľúče, preto je hash ku každému reťazcu deterministicky určený. Teda pre rovnaký vstup vyjde vždy rovnaký výstup. (Napr. MD5 hash k heslu admin bude vždy 21232f297a57a5a743894a0e4a801fc3.)

1.2 Cloud Computing

Cloud computing je sieť počítačov, v ktorej sú poskytované určité konkrétne služby. Výraz „cloud“ je už tradične zaužívaný ako metafora a pomáha abstraktne vnímať zložitosť tohto pojmu. Myšlienka je v tom, že hardvér a softvér môžu byť k dispozícii cez internet a tak naplniť potreby používateľov. Tieto výpočtové služby môžu byť teda poskytované aj na diaľku.

NIST (National Institute of Standards and Technology) definoval cloud computing v roku 2011 takto: „Cloud computing je model, ktorý umožňuje všadeprítomný, pohodlný a praktický prístup cez sieť na požiadanie k zdielanej skupine konfigurovateľných výpočtových zdrojov (napr. siete, servery, úložný priestor, aplikácie a služby), ktoré môžu byť rýchlo zabezpečené a vydané s minimálnym manažovaním alebo interakciou s poskytovateľmi služby. Tento model je zložený z 5 základných charakteristík, 3 modelov služieb a 4 modelov nasadení.“ [43] (obr. 1.1)



Obr. 1.1: NIST model cloud computingu. (Autor: Young-Chan Lee, Zdroj: `https://www.researchgate.net/publication/259972968_A_Deployment_Model_for_Cloud_Computing_using_the_Analytic_Hierarchy_Process_and_BCOR_Analysis`)

5 základných charakteristík cloud computingu.

- **On-demand self-service.** Samoobslužná služba na požiadanie. Používateľ môže jednostranne požiadať o zvýšenie alebo zníženie služby bez potreby ľudskej interakcie medzi používateľom a poskytovateľom cloudu. Používateľ teda môže požiadať o navýšenie úložného miesta alebo o viac virtuálnych strojov pre zvýšenie výpočtovej sily jednoducho a vtedy, keď potrebuje.

- **Broad network access.** Široký prístup. Zdroje sú k dispozícii prostredníctvom internetu a podporujú rôznorodých klientov (jednoduchých koncových používateľov aj multifukčné servery) aj rôzne platformy ako sú notebooky, smartfóny, tablety, pracovné stanice.

- **Resource pooling.** Rozdeľovanie zdrojov. Poskytovateľ cloude efetkívne rozdeľuje svoje zdroje. Združené zdroje slúžia viacerým používateľom a sú dynamicky prerozdeľované podľa ich potrieb. Toto vedie k tomu, že používateľ v danom okamihu nevie, kde presne sa fyzicky nachádzajú zdroje, ktoré mu boli poskytnuté.

- **Rapid elasticity.** Rýchla pružnosť. Kapacita a možnosti služieb sú pružné a vedia sa automaticky a rýchlo zmenšovať a zväčšovať podľa potrieb používateľa. Pre používateľa sa zdá ako keby boli služby neobmedzené a môžu mu byť kedykoľvek priradené v akomkoľvek množstve.

- **Measured service.** Meranie služby. Cloudové systémy monitorujú využívanie svojich zdrojov, pretože toto využívanie je veľmi dynamické. Z toho vedia následne vytvoriť prípadne poskytnúť prehľad o využívaní a podľa toho upraviť ceny služieb.

3 modely služieb.

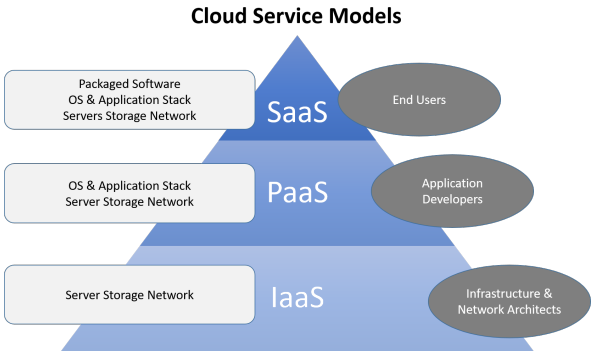
Cloudové služby môžu byť rozdelené do 3 základných modelov podľa ich levelu abstrakcie. (obr. 1.2)

- **Software as a service** - SaaS, Softvér ako služba. Poskytovateľ cloudovej služby umožňuje na svojich priestoroch prístup k aplikácií. Zákazník tejto služby je teda len koncový používateľ, len využíva aplikáciu, softvér, ktorý beží na cloude poskytovateľa. Používateľ nemôže konfigurovať infraštruktúru cloudu, nemôže modifikovať komponenty ako databáza, operačný systém a podobne. Takouto službou je napr. Gmail od Googlu.

- **Platform as a Service** - PaaS, Platforma ako služba. Služba poskytovaná pri PaaS zákazníkovi, je možnosť zákazníka umiestniť na cloud svoju aplikáciu, ktorá

používa nástroje, knižnice, programovacie jazyky, ktoré sú podporované poskytovateľom cloudu. Zákazník vďaka tomu vie ponúknuť svoju aplikáciu na používanie aj tretím stranám, pričom tretia strana nepotrebuje mať špeciálny hardvér či inštalovať špecifický softvér, ale využíva ju z cloudu. Zákazník ale opäť nemôže meniť a konfigurovať infraštruktúru cloudu.

- **Infrastructure as a Service** - IaaS, Infraštruktúra ako služba. Zákazník si od poskytovateľa tejto služby zapožičia infraštruktúru, ako hardvérovú, tak aj softvérovú. Zákazník tu môže následne napr. umiestniť ľubovoľný softvér vrátane operačného systému. Zákazník stále nemôže meniť základnú infraštruktúru cloudu poskytovateľa, ale má kontrolu nad operačným systémom, úložiskom, umiestnenými aplikáciami a čiastočne aj nad sieťovými komponentami, napr. firewallom.



Obr. 1.2: NIST modely služieb. (Zdroj: `https://medium.com/@IDMdatasecurity/types-of-cloud-services-b54e5b574f6`)

4 modely nasadenia.

Rozdelenie podľa prístupu používateľa.

- **Public cloud** Verejný cloud. Cloud a jeho súčasti sú poskytnuté na použitie širokej verejnosti. Neuplatňujú sa tu žiadne obmedzenia prístupu. Zväčša ponúkajú zákazníkom prístup k obrovským výpočtovým silám, keď práve zákazník potrebuje použiť na svoje potreby rýchlo veľké množstvo zdrojov (hardvéru či softvéru) a nemusí si tak kvôli aktuálnemu problému zabezpečovať sám požadované zdroje, ale využije hardvérové a softvérové možnosti nejakej cloudovej služby. Takýto cloud môže byť teda spravovaný a poskytnutý napr. školou - pre študentov alebo v rôznych iných organizáciách.

- **Private cloud** Súkromný cloud. Cloud je poskytnutý výlučne jednému zákazníkovi, jednej organizácii. Cloud je prispôsobený jeho požiadavkám na infraštruktúru.

Pre tento typ služby sa zvyčajne uzatvára aj zmluva. Tento model je preto vhodný pre použitie na spracovávanie osobných údajov.

- **Community cloud** Komunitný clolud. Tento typ nasadenia podporuje konkrétne zoskupenie ľudí alebo firiem s rovnakým zameraním a cieľom. Komunitný cloud preto spĺňa špecifické požiadavky nejakej skupiny, napr. požiadavky na bezpečnosť. Je podobný verejnému cloudu, avšak túto infraštruktúru môžu vlastniť aj jednotliví členovia komunity a mať za ňu zodpovednosť alebo tiež tretia strana a od spoločností vyberať poplatky za používanie. Taktiež je podobný súkromnému cloudu v tom, že takýto cloud je rovnako viac špecializovaný a používaný menšou skupinou ľudí a zväčša obmedzeným prístupom. Tu výhodou oproti súkromnému cloudu je menšia cena, pretože je zdielaná medzi viacerými subjektami.

- **Hybrid cloud** Hybridný cloud. Infraštruktúra cloudu je zložením 2 alebo viacerých odlišných infraštruktúr (z už spomenutých 3 modelov nasadenia.) Takýto spôsob využívania cloudu je veľmi praktický, ak má firma napr. zopár citlivých dát, ktoré vyžadujú vyššiu úroveň bezpečnosti, tieto uloží na súkromný cloud, zatiaľ čo pre ostatné svoje údaje môže použiť verejný cloud. Takto ich to bude stáť oveľa menej finančných nákladov ako keby používali na všetky svoje dáta súkromný cloud. Nevýhodou však môže byť náročnejšia manažovateľnosť.

Táto kapitola bola čerpaná z technickej správy NISTu o cloud computingu [43].

1.3 Používanie cloudu v súčastnosti

Viete koľko cloudových služieb využívate? V časopise Security World, v článku *SOC ve světě cloudů* [71] sa píše: „Zamerať sa len na niekoľko málo schválených cloudových služieb (a používať len tieto schválené - pozn. autorky práce) by mohla byť chyba. Mnoho bezpečnostných tímov ani netuší, koľko cloudových služieb ich firmy používajú.“ Ďalej je v tomto článku citovaný Gautam Kanaparthi, riaditeľ produktového manažmentu firmy Netskope, ktorá je dodávateľom zabezpečenia pre cloud, ktorý povedal: „Keď sa pýtam zákazníkov, koľko cloudových služieb používajú, odpovedajú 5, 10, možno 20. A keď urobíme kontrolu, nachádzame stovky cloudových aplikácií, v niektorých prípadoch sú ich dokonca aj tisíce“. Podľa najnovších správ Netskope o zabezpečení cloudu teraz priemerný podnik používa 1246 rôznych cloudových služieb, čo je nárast o 22% z 1022 služieb pred 1 rokom. To v priemere zahrňuje 175 rôznych služieb v oblasti ľudských zdrojov, 170 súvisiacich s marketingom, 110 pre spoluprácu a 76 v oblasti financií a CRM (riadenia vzťahov so zákazníkmi). **Ktorákoľvek z nich by potencionálne mohla pracovať s citlivými firemnými dátami a absolútna**

väčšina z nich nie je pripravená na podnikové využitie, tvrdí spoločnosť Netskope. ([71], 2019)

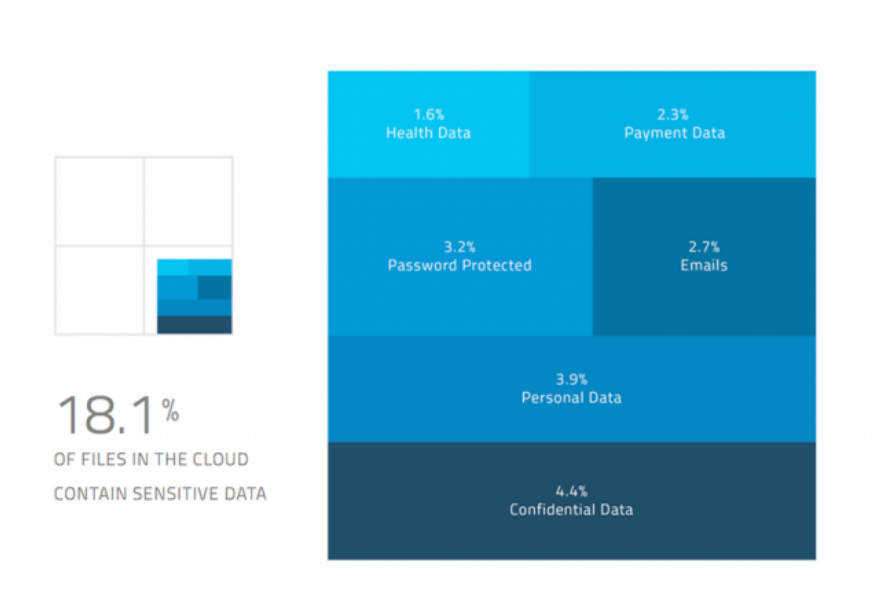
V poslednom období môžeme pozorovať veľký nárast využívania cloudových služieb, je to najmä vďaka mnohým výhodám, ktoré poskytujú, ako napríklad vysoká mobilita a možnosť prístupu k dátam a aplikáciám z rôznych častí Zeme, flexibilná škálovateľnosť a ušetrenie financií a pracovnej sily na hardvér a softvér a jeho správu a udržiavanie. Firma Gartner, Inc. je svetová výskumná a poradenská firma, ktorá poskytuje informácie, rady a nástroje v oblasti IT, financí atď. Táto firma okrem iného aj každoročne porovnáva nárast celkového príjmu z celosvetových IaaS Public cloud služieb. Celosvetový IaaS trh vzrástol o 32.8% v roku 2015 oproti roku 2014 [44], potom o ďalších 31% v roku 2016 [34], následne o 29,5% v roku 2017 [27] a ďalej o 31,3% v roku 2018 [26] (percentá sú vždy uvádzné vzhľadom na predchádzajúci rok k uvedenému roku). 5 najväščích IaaS poskytovateľov, ktorými sú (z roku 2018) Amazon (s podielom na trhu 47,8%), Microsoft (15,5%), Alibaba(7,7%), Google (4,0%) a IBM (1,8%), tvorí dokopy 77% IaaS trhu. Tieto firmy dosiahli nárast v roku 2018 oporti roku 2017 až 39%, zatiaľ čo ostatných 23% poskytovateľov len nárast o približne 11%. [26]

1.4 Citlivé údaje a cloud

Odborníci z firmy Skyhigh, ktorá sa zaoberá práve bezpečnosťou cloudu a ktorú odkúpila antivírusová a bezpečnostná firma McAfee, uverejnili na svojom blogu zaujímavý report. Na konci roka 2016 zanalyzovali využívanie cloudu viac ako 30 miliónmi zamestnancov vo viac ako 600 firmách po celom svete. Zistili nárast množstva citlivých údajov na cloude firmami. 18,1% všetkých dokumentov nahraných na cloud obsahovalo nejakú citlivú informáciu. Rozbité na konkrétnejšie kategórie to bolo:

- 4,4% dôverných dát (napr. finančné záznamy, obchodné plány, zdrojový kód, obchodné algoritmy)
- 3,9% dát obsahujúcich osobné identifikačné údaje (napr. dátumy narodenia, rodné čísla, telefónne čísla, daňové čísla)
- 3,2% dát chránených heslom, ktoré väčšina bezbečnostných riešení neprezerá (napr. heslom chránené ZIP súbory, Excel spreadsheets)
- 2,7% emailov (napr. exporty z Microsoft Outlook, súkromné EML a MSG správy)
- 2,3% dát obsahujúcich platobné údaje (napr. čísla kreditných kárt, debitných kárt, čísla bankových účtov)
- 1,6% dát obsahujúcich zdravotné údaje (napr. diagnózy pacienta, lekárske ošetrenia, ID čísla zdravotných záznamov)

Vizualizácia percent v obrázku 1.3. Taktiež uvádzajú, že pre zamestnancov nie je neobvyklé ukladať si svoje heslá vo wordovskom dokumente s názvom napríklad MyPasswords.docx a nahrať aj takýto súbor na zdielané médium. [17]



Obr. 1.3: Podiel citlivých údajov v cloude, Dec 2016. [17]

Ďalšie zaujímavé zistenia firmy skyhigh z roku 2017 [39]:

- Priemerná firma používa 1427 vzdialených cloudových služieb.
- 2,7% súborov zdielaných na cloude má prístupové práva také, že tieto súbory sú verejne dostupné.
- 71,3% všetkých cloudových služieb sú orientované na podniky a organizácie, 28,7% je orientovaných pre spotrebiteľov.
- Priemerná firma čelí 23,2 cloudovo orientovaným hrozbám za mesiac. Toto číslo narástlo za rok 2017 o 18,4%.
- Priemerná firma čelí 10,9 cloudovo orientovaným vnútorným hrozbám za mesiac. A až 93,5% organizácií čelí aspoň 1 takejto vnútornej hrozbe každý mesiac. (Vnútorné hrozby sú vlastní zamestnanci firmy, ktorí či už zlomyselne alebo z nedbalosti, neopatrnosti či nedostatočnej vzdelanosti urobia nevedome niečo zlé alebo hlúpe, čo môže otvoriť cestu aj útočníkom zvonka.)
- 31,3% zo všetkých cloudových služieb sú blokované vo firmách, pretože sú príliš riskantné.

1.5 Výhody cloudu

Firmy, ktoré adaptovali cloudové služby zaznamenali 20,66% priemerné zlepšenie v čase na obchodovanie, 18,8% priemerné zvýšenie efektvity spracovania, 15,07% priemerné zníženie nákladov na IT či 16,76% priemerné zníženie nákladov na správu IT, čo dokopy viedlo v priemere ku 19,63% -nému zvýšeniu rastu firmy [16]. Medzi 11 výhod cloudu, ktoré boli identifikované firmou Vanson Bourne patria:

- **Čerstvý softvér** - Ak sa používa cloud ako SaaS - softvér je prístupný hneď a po update je nová verzia softvéru hneď aktuálna pre všetkých používateľov.
- **Robiť viac s menej** - Firma môže vďaka cloudu redukovať veľkosť svojich datacentier alebo nemať žiadne ako aj znížiť počet zamestnancov na správu datacentier - výrazné zníženie nákladov na IT.
- **Flexibilná cena** - Firmy platia len za to, čo aktuálne potrebujú a používajú. Ak je práve vyťaženejšia doba, môžu si navýšiť zdroje a potom znovu znížiť ak ich už nepotrebujú. V bežnej firme by pre vyťaženejší čas museli kúpiť nové zdroje a potom ich pravdepodobne len nechať nečinné ak ich zrovna nepotrebujú.
- **Vždy prístupné** - Väčšina cloudových poskytovateľov je extrémne spoľahlivých v poskytovaní služieb a udržujú 99,99% dostupnosť. Ak má používateľ cloudu internetové pripojenie, môže pristupovať k jeho zdrojom prakticky odkiaľkoľvek.
- **Lepšia mobilita** - Zamestnancom sú ich dáta a aplikácie prístupné nech sa nachádzajú kdekoľvek, môžu pracovať cez telefón či tablet, môžu dokonca prísť kdekoľvek za zákazníkom a tam mu prezentovať ich firemné dáta a aplikácie.
- **Lepšia kolaborácia** - aplikácia cloudu tiež vylepšuje spoluprácu rozdielnych skupín ľudí a zdielanie ich práce v reálnom čase cez zdielané úložisko. Toto vylepšuje čas uvedenia na trh, vývoj produktov a zákaznícky servis.
- **Efektívnejšie využitie nákladov** - Firmy nemusia kupovať vybavenie a stavať a spravovať datacentrá. Bez clodu by mohli najskôr investovať milióny do kúpy hardvéru, priestorov a rôznych pomôcok bez dostania žiadnej rýchlej spätnej väzby z ich investície.
- **Náklady sa dajú rýchlo znížiť** - Počas času recesie alebo úpadku podnikania (ako je to v súčasnosti v energetickom priemysle) je tu možnosť flexibilnej štruktúry nákladov.
- **Flexibilná kapacita** - Cloud je veľmi flexibilný, v závislosti od okolností si môže firma zdroje navýšiť, znížiť alebo vypnúť úplne.

- **Uľahčenie fúzií a akvizícií** - Cloud umožňuje rýchlejšie zmeny, ak sa chcú napríklad 2 spoločnosti zlúčiť alebo spojiť, je to s cloudom oveľa rýchlejšie a efektívnejšie. Bez použitia cloudu by mohla migrácia aplikácií a vyradenie a zlúčenie data centier trvať roky kým by bežali obe spoločnosti na rovnkom IT stacku.

- **Menší dopad na životné prostredie** - S menším počtom datacentier vo svete a s efektívnejšimi operáciami máme kolektívne menší dopad na životné prostredie. Firmy, ktoré používajú zdielané zdroje, zlepšujú svoje „zelené“ kredity.

Napriek týmto výhodám tu sú aj mnohé bariéry, kvôli ktorým sú firmy zdržanlivé od adaptácie cloudu [16]. Tieto si uvedieme v nasledujúcej podkapitole.

1.6 Zábrany firiem voči cloudu

Skyhigh taktiež identifikoval hlavné dôvody, kvôli ktorým firmy ešte neprešli na používanie cloudu. Prieskum robili na viac ako 200 IT a IT Security lídroch firiem. Firmy sa boja o to, ako veľmi v bezpečí sú ich dáta v okamihu ako opustia firemný firewall. Až 73% respondentov uviedlo ako hlavný problém bezpečnosť dát, druhým dôvodom s počtom 38% opýtaných uviedlo ako dôvod nekompatibilitu so zákonmi a nariadeniami ich krajiny, ktoré môžu viesť k drahým pokutám a poplatkom. [20]

Ďalší ich report poukazuje na ďalšie problémy použitia cloudov na „tajné“ dáta firmy. Z viac ako 12000 cloudových služieb len 7% spĺňa požiadavky firiem, len 15,4% podporuje aj viac-faktorovú autentikáciu a len 9,4% z nich ukladá dáta šifrovane. Taktiež prišli z výsledkom, že 74,3% cloudových služieb nie je vhodných na uchovávanie európskych dát a je to práve preto, že 74,3% cloudových služieb neukladá dáta na území EÚ alebo na území štátu s ekvivalentne prísnymi zákonmi ochrany dát. [19] Na rovnakej skupine 12000 cloudov zistili, že 81,8% cloudových služieb prenáša dáta medzi používateľom a cloudom v zašifrovanej forme a to s pomocou SSL alebo TLS, avšak ako sme už spomínali, len 9,4% cloudov ukladá dáta u seba v šifrovanej forme. To znamená, že existuje aspoň 10000 cloudových služieb, ktoré ukladáju používateľské dáta v plaintexte, medzi nimi aj Gmail či Paypal. Priemerná firma nahrá na cloud približne 13,9 TB dát každý mesiac. 34% používateľov už niekedy nahrali na cloud nejakú citlivú informáciu ako napríklad osobné identifikačné údaje, zdravotné údaje, dáta s platobnej karty a iné. Taktiež napríklad v zákonoch USA je možnosť vlády vypýtať si od cloudového poskytovateľa ľubovoľné dáta bez toho, aby o tom oboznámili uživateľov cloudu. Najlepšou možnosťou ako sa ochrániť pred niečím takým je zašifrovať dáta v cloude s použitím a správou vlastných kľúčov, hoci len 1,1% cloudových poskytovateľov podporuje aj zákazníkom manažovateľné šifrovacie kľúče. [18]

1.7 Záver/Zhrnutie kapitoly

V tejto kapitole sme si rozobrali rôzne delenia cloudu podľa rozličných kritérií a na základe toho, čo poskytujú. Následne sme ukázali, že používanie cloudu je naozaj veľmi rozšírené, cloudové služby sú takmer všade aj keď ich nevnímame a jeho využívanie každým rokom stále nezanedbateľne stúpa. Medzi najväčšie výhody cloudu jednoznačne patrí fakt, že firma využívajúca cloud na určité dáta nemusí kvôli týmto dátam spravovať servery lokálne ako aj prístup k dátam z rôznych častí Zeme.

Ako ďalšie sme poukázali na fakt, že až približne pätina údajov na cloude sú práve citlivé a osobné údaje. To je popri desiatkovom počte cloudovo orientovaných hrozieb na jednu firmu mesačne vcelku desivé a malo by to zaujať našu pozornosť a obozretnosť.

Nakoniec sme zistili, že medzi hlavné dôvody, prečo aj napriek mnohým výhodám požívania cloudu firmy cloud nepoužívajú, patrí najmä strach o bezpečnosť takýchto dát a následne nekompatibilita cloudu so zákonmi krajiny.

Z tohto všetkého vyplýva, že téma ochrany údajov odosielaných na cloud by nám nemala byť ľahostajná a je potreba sa jej venovať.

Kapitola 2

Zákony o ochrane osobných údajov

Mnohé firmy a spoločnosti sa boja práve zákonov a pokút (kap. 1.6) a radšej spravujú údaje u seba interne na svojich serveroch a majú serverové miestnosti. Toto ich môže zbytočne zaťažovať finančne, priestorovo aj pracovnou silou - musia si napríklad zabezpečiť správcu serverov či technika v prípade poruchy, zabezpečenie miestnosti napríklad kamerami a zabezpečiť vlastné riešenie bezpečnosti prístupu k týmto dátam v rámci firmy. V tejto kapitole si rozoberieme problematiku ochrany osobných údajov a oboznámime sa s príslušnými zákonmi SR a GDPR v takej súvislosti a do takej miery, v akej to je pre potreby tejto práce nevyhnutné. V kapitole používame skratku OÚ pre výraz „osobné údaje“. Taktiež „Úrad na ochranu osobných údajov Slovenskej republiky“ budeme skrátene označovať slovom „úrad“.

Niektoré základné pojmy [1]:

- **Dotknutá osoba** - každá fyzická osoba, ktorej osobné údaje sa spracúvajú.
- **Prevádzkovateľ** - každý, kto vymedzí účel a prostriedky spracúvania osobných údajov a spracúva OÚ vo vlastnom mene.
- **Sprostredkovateľ** - každý, kto spracúva OÚ v mene prevádzkovateľa.
- **Tretia strana** - každý, kto nie je dotknutou osobou, prevádzkovateľom ani sprostredkovateľom alebo inou fyzickou osobou, ktorá na základe poverenia od prevázdkovateľa alebo sprostredkovateľa spracúva osobné údaje.
- **Spracúvanie osobných údajov** - je operácia alebo skupina operácií s osobnými údajmi, hlavne získavanie, zaznamenávanie, usporadúvanie, štruktúrovanie, uchovávanie, zmena, vyhľadávanie, prehliadanie, využívanie, poskytovanie prenosom, šírením alebo iným spôsobom, preskupovanie alebo kombinovanie, obmedzenie, vymazanie. Toto všetko bez ohľadu na to či sa to vykonáva automatizovanými alebo neautomatizovanými prostriedkami.

- **Šifrovanie** - transformácia osobných údajov takým spôsobom, po ktorom je opätovné spracúvanie možné len po zadaní správného parametra, ako je kľúč alebo heslo.

- **Tretia krajina** - krajina, ktorá nie je členským štátom EÚ.

2.1 Cloudové služby z pohľadu zákona o ochrane osobných údajov

Čerpané z Metodického usmernenia vydaného úradom na ochranu osobných údajov v SR [50]. Toto usmernenie vychádza z momentálne už neplatného zákona, avšak vzťahuje sa špeciálne na cloudové služby a s nimi spojenú ochranu osobných údajov.

1. Cloud ako služba.

Z hľadiska zákona č.122/2013 Z.z. o ochrane osobných údajov v znení neskorších predpisov zákona č.84/2014 Z.z. majú poskytovateľ cloudovej služby aj zákazník cloudovej služby veľa povinností vo vzťahu k dotknutým osobám. Tými sú napríklad uzavretie zmluvy, ktorá upravuje vzájomné práva a povinnosti, rôzne povinnosti vo vzťahu k dotknutým osobám či povinnosti týkajúce sa prenosu osobných údajov po siete cez určité územia. Aj keď cloud prináša v tejto oblasti nové možnosti, prináša aj veľa hrozieb z pohľadu ochrany osobných údajov, ako je najmä možná strata kontroly nad spracúvavanými osobnými údajmi a prístup iných strán k údajom. Táto hrozba tu je hlavne pre to, že poskytovateľ cloudovej služby používa vlastné mechanizmy, ako napr. technické a personálne opatrenia, prístup ďalších strán k údajom, a podobne.

2. Postavenie subjektov cloudu.

Postavenie zákazníka a poskytovateľa cloudovej služby. Zákazník cloudovej služby určuje konečný účel a predstavuje prevádzkovateľa. Prevádzkovateľ nesie prvotnú zodpovednosť za spracovanie osobných údajov v súlade so zákonom. Ako sprostredkovateľ vystupuje poskytovateľ cloudovej služby, na ktorého prevádzkovateľ prenáša niektoré povinnosti. Teda sprostredkovateľ spracúva osobné údaje v mene prevádzkovateľa, za podmienok dohodnutých písomne.

Postavenie subdodávateľov. Subdodávateľ je ďalší subjekt, ktorého sprostredkovateľ poveril spracúvavaním osobných údajov prevádzkovateľa , na základe písomnej dohody.

3. Zmluvné vzťahy.

Zmluva medzi zákazníkom a poskytovateľom cloudovej služby. Zákazník cloudovej služby a poskytovateľ cloudovej služby musia v súlade so zákonom o ochrane osobných údajov uzatvoriť písomnú zmluvu, ktorá musí obsahovať všetky náležitosti podľa § 8 ods. 4 zákona č.122/2013 Z.z.. Sprostredkovateľ musí spracúvať osobné údaje

len v rozsahu dohodnutých podmienok s prevádzkovateľom a nemôže počas účinnosti zmluvy jednostranne meniť podmienky.

Zmluvné postavenie subdodávateľov. Zodpovednosť za ochranu osobných údajov má sprostredkovateľ, aj keď ich ďalej spracúva subdodávateľ. Prevádzkovateľ však musí dať súhlas, že sprostredkovateľ bude spracúvať OÚ prostredníctvom inej osoby, subdodávateľa.

4. Cezhraničný prenos osobných údajov.

Pri cezhraničnom prenose osobných údajov sa uplatňujú predpisy krajiny, v ktorej sídli prevádzkovateľ cloudovej služby. Prenos osobných údajov môže byť v členských krajinách Europskej únie, komplikovanejší je prenos do tretej krajiny, alebo do krajiny nezaručujúcej primeranú úroveň ochrany osobných údajov. Pri poslednom uvedenom prenose je potrebné začleniť štandardné zmluvné doložky do zmluvy o cezhraničnom prenose podľa § 31 ods. 2 zákona č.122/2013 Z.z..

Záver tohto metodického usmernenia.

Používanie cloudu má mnohé výhody, ale prináša aj mnohé hrozby. Zákazníci cloudového systému by si mali premyslieť či budú túto službu používať a či dokáže cloud ich nároky a podmienky vyplniť. Výber vhodného poskytovateľa cloudu však nie je postačujúci, pre zabezpečenie ochrany osobných údajov je nutné chrániť a kontrolovať osobné údaje počas celého trvania ich spracúvania.

2.2 Zákony o ochrane osobných údajov SR

Nasledujúce údaje sú výňatky a citácie z GDPR a zo Zákona č. 18/2018 Z. z. [1].

Aktuálne platný zákon o ochrane osobných údajov u nás sa nazýva Zákon č. 18/2018 a nadobudol účinnosť dňa 25.5.2018. Tento zákon je v súlade s **Nadriadením** (GDPR) Európskeho parlamentu a rady (EÚ) č.2016/679 o ochrane fyzických osôb pri spracúvaní osobných údajov a o voľnom pohybe takýchto údajov a **Smernicou** Európskeho parlamentu a rady (EÚ) č.2016/680 o ochrane fyzických osôb pri spracúvaní osobných údajov príslušnými orgánmi na účely predchádzania trestným činom, ich vyšetrovania, odhaľovania alebo stíhania alebo na účely výkonu trestných sankcií a o voľnom pohybe takýchto údajov. Zákon sa podľa §3 vzťahuje na spracúvanie osobných údajov v rámci činnosti prevádzkovateľa alebo sprostredkovateľa umiestneného v SR ako aj na spracúvanie osobných údajov dotknutej osoby, ktorá sa nachádza na území SR (aj prevádzkovateľom alebo sprostredkovateľom neumiestnenom v žiadnom členskom štáte).

Osobným údajom sa rozumie každý údaj, ktorý sa týka už identifikovanej osoby a každý údaj, podľa ktorého je fyzická osoba identifikovateľná či už priamo alebo nepriamo. Takýto identifikátor môže byť napríklad meno, priezvisko, lokalizačné údaje alebo

aj (jedna alebo viaceré) charakteristiky alebo znaky osoby, ktoré tvoria osobe fyzickú identitu, fyziologickú identitu, genetickú identitu, psychickú identitu, mentálnu identitu, ekonomickú identitu, kultúrnu identitu alebo sociálnu identitu (čiže aj napríklad oslovenie „mama“ je takýto osobný údaj, lebo dieťa vie podľa neho identifikovať mamu).

Ďalej sa pozrieme na vybrané časti z práv a povinností oboch strán.

2.2.1 Zásady spracúvania osobných údajov

Prevádzkovateľ má zo zákona ([1], §12 **Zásada zodpovednosti**) zodpovednosť dodržiavať základné zásady spracúvania OÚ, zabezpečiť súlad spracúvania osobných údajov so zásadmi spracúvania osobných údajov a taktiež má v prípade požiadania úradu povinnosť tento súlad preukázať.

Medzi tieto zásady, okrem iného patrí aj **Zásada správnosti** ([1], §9), ktorá hovorí, že osobné údaje, ktoré spracúva prevádzkovateľ musia byť správne a v prípade ich zmeny príslušne aktualizované. Prevádzkovateľ tiež musí prijať potrebné účinné opatrenia na to, aby sa takéto nesprávne osobné údaje bez zbytočného odkladu odstránili alebo opravili, prípadne označili, že sú nesprávne. Ďalšou zásadou je **Zásada integrity a dôvernosti** ([1], §11), podľa ktorej osobné údaje musia byť spracúvané takým spôsobom, aby bola pomocou primeraných technických a organizačných opartení zabezpečená primeraná bezpečnosťosobných údajov. Taktiež musia byť takto zabezpečené pred neoprávneným alebo nezákonným spracúvaním osobných údajov, pred náhodnou stratou, vymazaním alebo poškodením osobných údajov.

Ďalšou podmienkou pre prevádzkovateľa je, že ak firma má a spracúva osobné údaje dotknutej osoby, ktoré je založené na súhlase tejto osoby, prevádzkovateľ má povinnosť byť kedykoľvek schopný preukázať, že požadovaný súhlas so spracovaním osobných údajov mu dotknutá osoba naozaj poskytla ([1], **§14**).

Ako poslednú časť z tohto úseku zákona uvedieme **Právo na prístup k osobným údajom** ([1], §21). Dotknutá osoba má z tohto zákona právo kedykoľvek požiadať prevádzkovateľa, ktorý o nej spracúva osobné údaje o osobné údaje, ktoré o nej spracúva a prevádzkovateľ je povinný jej ich poskytnúť, ako aj informácie o účele ich spracúvania, kategórii, dobe uchovávania, zdroji či informácie o tom, komu ich prevádzkovateľ poskytol. Prvé takéto požiadanie by malo byť bezplatné, pri opakovanom žiadaní tej istej dotknutej osoby môže prevádzkovateľ požadovať od dotknutej osoby financie na pokrytie administratívnych nakladov na tento úkon.

2.2.2 Oprava a vymazanie a obmedzenie spracúvania osobných údajov

V tejto časti zákona si povieme niektoré ďalšie z práv dotknutej osoby týkajúce sa korekcie a redukcie spracovávaných osobných údajov.
Z **práva na opravu osobných údajov** ([1], §22) sa dozvedáme, že osoba, o ktorej sa spracuvávajú osobné údaje má právo na to, aby nesprávne údaje, ktoré sú o nej u prevádzkovateľa uložené, boli bez odkladu opravené prevádzkovateľom. Taktiež má právo na to, aby neúplné osobné údaje prevádzkovateľ doplnil. Dotknutá osoba má taktiež **Právo na výmaz osobných údajov** ([1], §23) za určitých podmienok. V tomto prípade musia byť osobné údaje, ktoré sa týkajú tejto dotknutej osoby bezodkladne vymazané. Takýmito dôvodmi na oprávnený výmaz osobných údajov je napríklad fakt, že osobné údaje už nie sú potrebné na ten zámer, na ktorý sa získali alebo tiež napríklad ak údaje boli získané na základe súhlasu dotknutej osoby a táto osoba svôj súhlas odvolala a nie je iné zákonné právo pre spracovávanie týchto osobných údajov. Tak isto má dotknutá osoba **Právo na obmedzenie spracúvania osobných údajov** ([1], §24) ak je napríklad takéto spracúvanie protizákonné, ale dotknutá osoba nesúhlasí s vymazaním jej osobných údajov a požaduje len ich obmedzenie alebo môže dotknutá osoba žiadať o obmedzenie spracúvania jej osobných údajov ak napríklad pre prevádzkovateľa už nie sú potrebné na zámer, na ktorý boli získané, ale sú potrebné aby si dotknutá mohla uplatniť nejaký právny nárok.

2.2.3 Právo na prenosnosť

Podľa §26 **Právo na prenosnosť osobných údajov** ([1], §26) má dotknutá osoba nárok na získanie svojich osobných údajov od prevádzkovateľa, ktoré mu poskytla a to v štruktúrovanom, bežne používanom a strojovo čitateľnom formáte. Tiež má právo na prenos týchto údajov inému prevádzkovateľovi, ak je to technicky možné.

2.2.4 Všeobecné povinnosti prevádzkovateľa a sprostredkovateľa

V tejto časti zákona sa dozvedáme o zabezpečení a ochrane osobných údajov.

Podľa §31 má **Prevádzkovateľ** ([1], §31) povinnosť schváliť adekvátne technické a organizačné opatrenia podľa povahy, rozsahu, účelu spracovania osobných údajov, zohľadňujúc hrozby a rozlišné pravdepodobnosti a závažnosti hrozieb vzhľadom na práva dotknutej osoby, ktorými zabezpečí a následne bude vedieť preukázať, že osobné údaje, ktoré spracuváva sú spracovávané v súlade s týmto zákonom. Zároveň má prevádzkovateľ povinnosť tieto opatrenia a zabezpečenia obnovovať podľa potreby. Taktiež je prevádzkovateľ povinný systematicky zisťovať či účel, na ktorý boli osobné údaje získané, stále trvá a v prípade konca trvania tohto účelu bezodkladne vymazať tieto osobné

údaje. Na dokázanie, že prevádzkovateľ splnil tieto povinnosti môže slúžiť schválený kódex správania alebo certifikát.

Ďalším paragrafom je paragraf s názvom **Špecificky navrhnutá a štandardná ochrana osobných údajov** ([1], §32) , ktorý má 4 časti. Tento paragraf pojednáva o nasledujúcich bodoch:

- Pred tým ako prevádzkovateľ začne spracúvať osobné údaje musí navrhnúť a začať používať a následne počas spracovávania osobných údajov používať špecificky navrhnuté zabezpečenie ochrany osobných údajov. Táto navrhnutá ochrana osobných údajov musí pozostávať najmä z prijatia patričných technických a organizačných opatrení a to hlavne pomocou pseudonymizácie. Týmito opatreniami sa má zabezpečiť dodržiavanie základných zásad uvedených v kapitole 2.2.1 a ostatných zásad, ktoré sme neuviedli, lebo sme ich nepovažovali za relevantné k tejto práci (môžte ich nájsť v [1], § 6 až § 12); a sú použité ako účinné zabezpečenie zodpovedajúcich záruk ochrany osobných údajov.

- Pri navrhovaní konkrétneho špecifického zabezpečenia ochrany osobných údajov podľa predchádzajúceho bodu musí prevádzkovateľ zobrať do úvahy najaktuálnejšie vedomosti a poznatky ochrany osobných údajov, taktiež musí zobrať do úvahy náklady na vykonanie takýchto oprávnení, povahu, rozsah, kontext a zámer spracúvania osobných údajov a rovnako ako v §31 aj hrozby a pravdepodobnosti naplnenia týchto hrozieb spolu s ich závažnosťou pre ochranu práv dotknutej osoby.

- Prevádzkovazeľ musí zabezpečiť požadovanú štandardnú ochranu osobných údajov tým, že prijme dostatočné technické a organizačné opatrenia na to, aby zabezpečil spracovávanie osobných údajov **len** na konkrétny účel, aby čo najviac obmedzil množstvo získaných osobných údajov, rozsah ich spracovávania, dobu ich uchovávania a dostupnosť takýchto osobných údajov. Taktiež sa prevádzkovateľ **musí postarať o to, aby bez zásahu inej fyzickej osoby neboli tieto osobné údaje bežne dostupné** ľubovoľnému počtu fyzických osôb.

- Na dokázanie, že prevádzkovateľ splnil tieto povinnosti môže slúžiť certifikát podľa [1] § 86.

Z ďalšieho paragrafu §34 - **Sprostredkovateľ** ([1], §34) - sa môžme dozvedieť, že ak chce prevádzkovateľ poveriť spracovávaním osobných údajov sprostredkovateľa, nepotrebuje na to súhlas osôb, o ktorých tieto údaje spracuváva, a teda dotknutá osoba nemusí vedieť či prevádzkovateľ využíva na spracovúvanie jej osobných údajov aj nejakú tretiu stranu. Prevádzkovateľ však takto môže poveriť len takého sprostredkovateľa, ktorý ponúka postačujúcu garanciu adekvátnych technických a organizačných opatrení

na to, aby narábanie s osobnými údajmi bolo v súlade s týmito zákonmi a aby bola zaručená ochrana práv dotknutých osôb.

2.2.5 Bezpečnosť osobných údajov

Ďalšia časť tohto zákona, bezpečnosť osobných údajov, hovorí o povinnostiach prevádzkovateľa a sprostredkovateľa vzhľadom na bezpečnosť osobných údajov. Paragraf zákona s názvom **Bezpečnosť spracúvania** ([1], §39) opäť dáva prevádzkovateľovi aj sprostredkovateľovi, ktorí spracúvajú osobné údaje, za povinnosť prijať a vykonať dostatočné technické a organizačné opatrenia so zreteľom na najaktuálnejšie vedomosti a poznatky z tejto oblasti a opäť musia prevádzkovateľ aj sprostredkovateľ zobrať do úvahy náklady na vykonanie takýchto oprávnení, povahu, rozsah, kontext a zámer spracúvania osobných údajov a aj hrozieb a pravdepodobnosti týchto hrozieb spolu s ich závažnosťou pre ochranu práv dotknutej osoby, pričom tieto opatrenia by mohli zahŕňať hlavne:

- pseudonymizáciu a šifrovanie osobných údajov,
- zabezpečenie trvalej dôvernosti, integrity a dostupnosti osobných údajov a dostupnosti a odolnosti systémov spracúvajúcich osobné údaje,
- obnovenie dostupnosti osobných údajov a zabezpečenie prístupu k osobným údajom ak nastane technický alebo fyzický incident,
- pravidelné testovanie, hodnotenie a posudzovanie účinnosti technických a organizačných opatrení na zabezpečenie ochrany osobných údajov pri ich spracúvaní.

Pri zvažovaní dostatočnosti úrovne bezpečnosti sa berú do úvahy hrozby a riziká spojené so spracovávaním osobných údajov, hlavne náhodné alebo neoprávnené a úmyselné zničenie dát, ich strata alebo zmena. Ako hrozba sa berie do úvahy aj neoprávnený prístup a protizákonné poskytovanie týchto osobných údajov či už pri prenášaní, uchovávaní alebo spracovávaním.

Na dokázanie splnenia uvedených 4 opatrení môže slúžiť schválený kódex správania podľa [1] § 85 alebo certifikát podľa [1] § 86.

Dôležitou súčasťou tohto paragrafu zákona je aj povinnosť prevádzkovateľa a rovnako aj sprostredkovateľa dôkladne poučiť a zabezpečiť, aby všetky fyzické osoby, ktoré k takýmto osobným údajom môžu pristupovať a narábať s nimi, tak robili len podľa pokynov prevádzkovateľa alebo podľa špeciálneho predpisu alebo podľa medzinárodnej zmluvy.

Oznámenie porušenia ochrany osobných údajov dotknutej osobe ([1], §41) hovorí o tom, že prevádzkovateľ musí v prípade narušenia bezpečnosti jej osobných údajov o tom bezodkladne informovať danú dotkntú osobu ak takéto narušenie môže viesť

aj k narušeniu práv dotknutej fyzickej osoby. Toto informovanie musí byť jednodché a jasné a vysvetlovať povahu narušenia, kontaktné údaje na získanie viacerých informácií, rozobratie očakávateľných následkov tohto narušenia a popis opatrení, ktoré prevádzkovateľ odsúhlasil alebo navrhol ako nápravu narušenia bezpečnosti, a opatrenia, ktoré odsúhlasil alebo navrhol aj voči možným následkom. Takéto uvedené oznámenie **nie je potrebné** ak:

- prevádzkovateľ zaviedol dostatočné technické a organizačné opatrenia na zabezpečenie ochrany tých osobných údajov, o ktoré sa jedná a u ktorých bola narušená ich bezpečnosť, a to hlavne šifrovaním alebo inými opatreniami, vďaka ktorým znečitatelnil tieto osobné údaje osobám, ktoré by k nim nemali mať právo pristupovať,
- prevádzkovateľ už schválil a uplatnil kroky k tomu, aby daný incident neohrozoval práva dotknutých fyzických osôb a neviedol k porušeniu týchto práv.
- by na to bola potrebná neadekvátna námaha, v tomto prípapde ale prevádzkovateľ musí informovať o tomto incidente verejnosť alebo hocijako inak dosiahnuť to, aby dotknuté osoby boli o tom oboznámené.

Ak prišlo k narušeniu bezpečnosti osobných údajov a predvádzkovateľ o tom stále ešte neinformoval doknuté osoby, môže po preskúmaní hrozieb a závažnosti úradom buď dostať žiadosť, aby tak spravil alebo úrad rozhodne, že bola splnená niektorá z vyššie uvedených podmienok.

2.2.6 Prenos OÚ do tretej krajiny alebo medzinárodnej organizácie

Prenos osobných údajov do tretej krajiny alebo medzinárodnej organizácii ([1], §48) hovorí o tom, že osobné údaje môžme preniesť do tretej krajiny v tom prípade, že Komisia už určila, že daná tretia krajina, kokrétny územný celok, dané odvetvie v tretej krajiny či medzinárodná organizácia dodržiavajú dostatočnú bezpečnosť v ohľade osobných údajov. V takomto prípade nie je potrebné osobitné povolenie. Druhou možnosťou na takýto prenos (ak nie je vydané takéto prehlásenie Komisie o tretej krajine) je, keď je zaručená garancia primeranej ochrany osobných údajov. Táto primeraná garancia sa dá stanoviť na základe medzinárodnej zmluvy, vnútropodnikových pravidiel, štandardnej doložky o ochrane údajov (ktorá bola prijatá Komisiou alebo úradom), schváleného kódexu správania alebo certifikátu. Tiež sa primeraná garancia ochrany dá zaručiť na základe povolenia úradom, a to zmluvnými doložkami medzi jednotlivými stranami alebo nariadeniami v administratívnych dohodách.

2.2.7 Práva a povinnosti príslušného orgánu a sprostredkovateľa

Príslušný orgán je napríklad orgán činný v trestnom konaní a takýto orgán môže získavať osobné údaje napríklad na účely trestného konania. Na takéto orgány sa vzťahuje okrem iného aj §69 - **Vedenie logov** ([1], §69). Takýto orgán musí mať vo svojom systéme automatické vytváranie logov o každom zisku, zmene, prezeraní, vymazaní či poskytnutí osobných údajov. Logy musia obsahovať informácie o dôvode, dátume a čase prezerania, identifikačné údaje osoby, ktorá si tieto údaje prezerala a v prípade poskytovania údajov ďalším osobám aj identitu týchto osôb. Príslušný orgán smie použiť tieto logy len na overenie toho či boli osobné údaje spracovávané podľa zákonov, na vlastné monitorovanie, na zabezpečenie integrity a bezpečnosti a pre potreby trestného konania. Príslušný orgán alebo jeho sprostredkovateľ musí na vyžiadanie dodať tieto logy úradu.

2.2.8 Správne delikty

Podľa §104 **Správne delikty** ([1], §104) môže prevádzkovateľ alebo aj sprostredkovateľ dostať pokutu úradom:

- až do 10.000.000 eur. V prípade podniku to môže byť pokuta do 2% celkového ročného obratu alebo do 10.000.000 eur - zvolí sa väčšia čiastka. Táto pokuta môže byť udelená napríklad za porušenie alebo nesplnenie niektorého zo zákonov v kapitole 2.2.4 alebo 2.2.5, ktoré sa týkajú zabezpečenia ochrany a bezpečnosti osobných údajov, či za porušenie nesplnenia povinnosti vedenia logov pre príslušné orgány z kapitoly 2.2.7.

- až do 20.000.000 eur. V prípade podniku to môže byť pokuta do 4% celkového ročného obratu alebo do 20.000.000 eur - zvolí sa väčšia čiastka. Takáto pokuta môže byť udelená napríklad za porušenie alebo nesplnenie niektorého zo zákonov uvedených v čati 2.2.1 - zásady spracúvania osobných údajov, pri nenaplnení práv dotknutej osoby na opravu, výmaz, obmedzenie či prenosnosť jej osobných údajov uvedených v kapitolách 2.2.2 a 2.2.3, ako aj za porušenie alebo nesplnenie niektorej z povinností pri prenose údajov do tretej krajiny z kapitoly 2.2.6.

Tieto pokuty môžu byť uložené aj orgánom verejnej moci či verejnoprávnym inštitúciám ak vystupujú v roli prevádzkovateľov.

Slovensko prevzalo z tohto zákona plnú výšku sumy pokuty ako jedna z mála krajín, v Maďarsku napr. zo začiatku dávali za prvý takýto priestupok len napomenutie.

2.3 Záver/Zhrnutie kapitoly

V tejto kapitole sme si najskôr popísali metodické usmernenie vydané úradom na ochranu osobných údajov. Ďalej sme prešli cez relevantné zákony k našej téme a dozvedeli sme sa povinnosti ľudí, ktorí osobné údaje spravujú a spracúvajú, že musia udržiavať takéto údaje správne, musia zachovať ich integritu a dôvernosť. Tiež musia klásť dôraz na ochranu ich bezpečnosti.

Taktiež sme si popísali rôzne práva osoby, o ktorej sú spracovávané osobné údaje. Takáto dotknutá osoba má právo napr. zísakť spracovávané údaje o sebe, má právo na to, aby boli údaje o nej vždy správne a aktualizované (napr. pri zmene priezviska, bydliska a podobne). Toto bolo v kapitole spomenuté na to, aby sme poukázali na fakt, že okrem ukladania údajov na cloud musíme zo zákona vedieť rôzne manipulovať s týmito dátami a vykonávať nad nimi aj špecifické databázové dotazy (query) a podľa možnosti to robiť v našom záujme aj efektívne.

Pokuty za nedodržanie zákonov sú vysoké, preto by malo byť pre firmu žiadané ich dodržiavať a osobné údaje patrične chrániť.

Kapitola 3

Existujúce techniky

V tejto kapitole si povieme niečo o už existujúcich technikách riešiacich problém ochrany a efektívneho prístupu údajov v cloudových službách a pojednáme o ich nedostatok či nevýhodách použitia. Týmito sú: Homomorfické šifrovanie, modely CryptDB, C-SDA, GhostDB a Helib. V práci používame slová homomorfický a homomorfný ako synonymá.

3.1 Homomorfické šifrovanie

3.1.1 Homomorfizmus

Homomorfizmus grúp sú zobrazenia medzi grupami, ktoré zachovávajú grupové operácie.

Definícia 3.1. Nech $(G, \odot)$, $(H, \star)$ sú grupy. Potom zobrazenie $f : G \to H$ je homomorfizmus, ak

$$f(g_1 \odot g_2) = f(g_1) \star f(g_2)$$

platí pre ľubovoľné $g_1, g_2 \in G$

My budeme používať len jedno „univerzum“ a teda zobrazenie $f : G \to G$. Nech $(G, \odot)$ a overíme či $f(g_1 \odot g_2) = f(g_1) \odot f(g_2)$

Príklady takýchto homomorfizmov:

Príklad 3.2. $(R, *) : f(x) = x^2$

Príklad 3.3. $(Z, +) : f(x) = 3x$

Príklad 3.4. $(Z \mod (p), +) : f(x) = x + p$

3.1.2 Homomorfický kryptosystém

Použitie homomorfického šifrovania v cloudových službách nám prinesie ako výsledok taký systém, kde pre prístup alebo modifikáciu dát nebudeme musieť sťahovať z cloudu

celé zašifrované dáta a dešifrovávať ich, ale operácie a príkazy sa vykonajú priamo na zašifrovaných dátach bez nutnosti ich dešifrácie.

Definícia 3.5. Homomorfický kryptosystém podľa [55].
Nech množina všetkých správ $(M, \odot)$ je konečná (semi-)grupa, a nech $\sigma \in N$ je bezpečnostný parameter. Homomorfická public-key šifrovacia schéma (alebo homomorfický kryptosystém) na **M** je štvorica (K, E, D, A) pravdepodobnostného, očakávane polynomiálneho časového algoritmu, spĺňajúca nasledujúce vlastnosti:

- **Generovanie kľúča:** Na vstupe 1^σ algoritmus dá na výstup šifrovací/dešifrovací pár kľúčov $(k_e, k_d) = k \in K$, kde K označuje množinu kľúčov.
- **Šifrovanie:** Na vstupe 1^σ, k_e, a prvku $m \in M$, šifrovací algoritmus E dá na výstup šifrovaný text $c \in C$, kde C označuje množinu šifrových textov.
- **Dešifrovanie:** Dešifrovací algoritmus D je deterministický. Na vstupe 1^σ, k, a prvku $c \in C$, dá na výstup prvok z množiny správ M taký, že pre všetky správy $m \in M$ platí: Ak $c = E(1^\sigma, k_e, m)$, potom $Pravdepodobnosť[D(1^\sigma, k, c) \neq m]$ je zanedbateľná, t.j. platí, že $Pravdepodobnosť[D(1^\sigma, k, c) \neq m] \leq 2^{-\sigma}$.
- **Homomofrická vlastnosť:** A je algoritmus ktorý na vstupe 1^σ, k_e, a prvkoch $c_1, c_2 \in C$ dá na výstup prvok $c_3 \in C$ taký, že pre všetky správy $m_1, m_2 \in M$ platí: Ak $m_3 = m_1 \odot m_2$ a $c_1 = E(1^\sigma, k_e, m_1)$ a $c_2 = E(1^\sigma, k_e, m_2)$, potom $Pravdepodobnosť[D(A(1^\sigma, k_e, c_1, c_2))] \neq m_3$ je zanedbateľná.

Homomorfický kryptosystém môže byť rozdelený na čiastočne homomorfný a (plne) homomorfný (FHE - fully homomorphic encryption). Čiastočne homomorfný systém môže byť **Aditívny** - zachováva vlastnosť pre +, ale nie pre *, a nazýva sa aditívny homomorfický kryptosystém. **Multiplikatívny** - zachováva vlastnosť pre krát, ale nie pre plus, nazýva sa multiplikatívny homomorfický kryptosystém. Takýmito čiastočne homomorfnými kryptosystémami sú napríklad aj RSA a ElGamal, ktoré oba podporujú len operácie násobenia [49].

Plne homomorfný kryptosystém zachováva operácie plus aj krát.

Prvá myšlienka pre vznik FHE a jej výhody boli publikované v roku 1978. Tri dekády sa nevedelo či je FHE vôbec teoreticky či prakticky možná. Neexistovala žiadna schéma, ktorá by povolovala vykonávať neohraničený počet sčítaní a násobení na zašifrovaných dátach zatiaľ čo by stále zachovávala homomorfickú vlastnosť. Prvým priblížením bola Boneho schéma, ktorá umožňovala ľubovoľný počet sčítaní, avšak len jedno násobenie.

Gentryho schéma

Ako prvý publikoval plne homomorfnú schému Craig Gentry, v roku 2009. Táto schéma podporovala neobmedzené množstvo sčítaní aj násobení na zašifrovaných dátach. Bola založená na predchádzajúcich schémach, ktoré podporovali len obmedzené počty použitia sčítania či násobenia a na neobmedzený počet týchto operácií bola rozšírená pomocou bootstraping-u. Problém bol v tom, že každá ďalšia operácia na zašifrovanom texte pridávala do tohto textu stále viac a viac šumu, čím sa šifrový text stal v jednej chvíli už nedešifrovateľným. Gentry bootstrapping-om vedel zredukovať mieru šumu a dokázal, že túto redukciu šumu vie vždy spraviť tak, aby bolo možné vykonať ešte aspoň jednu ďalšiu operáciu. Tým vie zabezpečiť neobmedzené množstvo použitia operácií. [28]

Gentry pracoval aj ďalej na tomto systéme a spolu s ďalšími vedcami sa mu podarilo v ďalších rokoch vyvinúť schému, ktorá bootstrapping nepotrebuje. Je založená na LWE (learning with error) - učení s chybami. Viac v práci Gentryho z roku 2012 [11] a následné vylepšenie tohto systému z roku 2014 [12].

Bližší prehľad o vývoji a rôznych verziách, smeroch, spôsoboch implementovanie, pokusoch a vylepšeniach homomorfického šifrovania s odkazmi na ďalšie zdroje si môžete pozrieť v práci Mohameda Alloghaniho a spoluatorov - A systematic review on the status and progress of homomorphic encryption technologies [4].

3.1.3 Praktické použitie homomorfického šifrovania v cloude

Homomorfické šifrovanie nám umožňuje mať výhody lokálneho šifrovania a zároveň vykonávať naše požadované operácie a databázové dotazy na serveri bez toho, aby ich server vedel odšifrovať. Uvedené na príklade:

Príklad 3.6. Zašifrujeme číslo 5 a zašifrujeme číslo 24 a sčítame na zašifrovaných hodnotách. Odšifrovaný výsledok nám dá čislo 29. Platí teda:

$$E(5) + E(24) = E(29)$$

My pracujeme ale nad univerzom ($Z \mod 2$), čo je množina všetkých bitov $\{0, 1\}$. Pracujeme nad homomorfizmom ($Z \mod 2, +$) a ($Z \mod 2, *$), čo znamená, že máme plne homomorfný kryptosystém. Nech KeyGen() je funkcia na generovanie kľúča, D(key, c) je funkcia na odšifrovanie a $E(key, message)$ je šifrujúca funkcia, ktoré spĺňajú homomorfickú vlastnosť. To znamená, že platí:

$$E(key, M_1 * M_2) = E(key, M_1) * E(key, M_2)$$

$$E(key, M_1 + M_2) = E(key, M_1) + E(key, M_2)$$

Ďalej, keďže E je homomorfická, môžme vytvoriť nasledovnú rovnicu:

$$E(key, 1 + a * b) = E(key, 1) + E(key, a) * E(key, b)$$

a	b	(1 + a * b)
0	0	1
0	1	1
1	0	1
1	1	0

Tabuľka 3.1: Tabuľka pravdivostných hodnôt pri homomorfickej rovnici.

Z tejto rovnice dosadíme hodnoty do tabuľky pravdivostných hodnôt:

Z výsledných bitov na pravej strane môžme spozorovať, že to, čo takto dostaneme, je funkcia zhodná svojimi hodnotami funkcii NAND. Vieme teda spraviť funkciu NAND a z tejto funkcie vieme vytvoriť aj všetky ostatné logické spojky, a teda aj ľubovoľnú Booleovskú funkciu, a tak teoreticky aj ľubovoľnú funkciu a vytvoriť ľubovoľný program a naopak, každý program vieme zredukovať až na logické hradlá. Teraz teda môžme použitím NAND hradiel vytvoriť deterministický program na základe homomorfických vlastností.

Nech f(x) = Evaluate(key, funkcia) je funkcia, ktorá zoberie kľúč a naprogramovanú funkciu a pomocou kľúča premení našu funkciu na sériu logických NAND hradiel, pošle ju na cloud. Cloud následne aplikuje túto funkciu na dáta/dokument uložené na cloude a vráti výstup naspäť. Používateľ sa po odšifrovaní dozvie výstup funkcie: $výsledok = D(key, f(x))$. Cloud počas celého behu nevie odšifrovať dáta a nepozná ani funkciu, ktorú má aplikovať a netuší ani čo funkcia robí.

Funkcie, ktoré takto vykonávame, musia byť „čisté funkcie“. To sú také, ktoré v prostredí, v ktorom sú, nič nemenia a len dajú výstup.

Problémom teraz nastáva fakt, že dáta sú meniteľné a hocikto môže poslať ľubovoľnú funkciu a tak dáta meniť, preto chceme zabezpečiť aj integritu. Programov na zabezpečenie integrity je veľa, jedným z nich je aj napr. HMAC (HMAC bližšie popisujeme v kap. 4.4.2). Tento HMAC je opäť len program, ktorý vie byť rozložiteľný na sériu NAND hradiel. Takže opäť len tento program napíšeme ako sériu NAND hradiel a pošleme na cloud.

Toto riešenie výhodne využíva cloud nielen ako úložisko dát, ale zároveň aj ako výpočtovú silu, keď celý beh funkcie prebieha na serveri. [29]

3.1.4 Nevýhody HE

Homomorfické šifrovanie je teda zjednodušene popísané šifrovacou shcémou, v ktorej operácie na 2 šifrovaných textoch dá za výsledok šifrovaný text, ktorý po odšifrovaní je rovnaký ako keby operácie boli vykonávané na plaintexte. Nevýhodou tohto šifrovania je ale efektívnosť. Na vykonanie jedného dotazu musí algoritmus v každom prípade

prejsť celou databázou. Prvý návrh algoritmu firmou IBM, pri ktorom pomáhal aj samotný Craig Gentry, vykonával operácie až 100 bilión-krát (10^{14}) pomalšie ako by trvali tieto operácie na plaintexte [24] [15] [6]. V roku 2013 prišlo zlepšenie na približne milión hlavne vďaka tomu, že použili 16 jadrový server. V tomto prípade teda, ak by trvala operácia na plaintexte 1 sekundu, tak pri homomorfickom šifrovaní by trvala až približne 12 dní. [13] V roku 2018 prišlo ďalej zlepšenie 15 až 75-krát. [15] Homomorfické šifrovanie je však stále veľmi pomalé a nepraktické v dnešnom uponáhľanom svete, kde aj čakanie na výsledok z databázy 2 sekundy je takmer neakceptovateľné používateľmi. Jeho ďalšou nevýhodou je aj nepoužiteľnosť pre viacerých používateľov a možnosť prístupu viacerých používateľov. [6]

Iným problémom - pri posielaní na cloud, môže byť veľkosť takýchto programov zložených len z NAND operácií. Tvorca popísaného systému skúšal naprogramovať niečo podobné ako program na hľadanie reťazca v texte. Tento kód mal okolo 3 TB. Medzi dohady na vylepšenie homomorfického šifrovania patria: rýchlejšie PC a komponenty, zmena žiadaných operácii (nie + a *, ale iné operácie), vytvorenie iných logických spojok (nie NAND), lepšia optimalizácia kódov zložených z NAND hradiel a podobne. [29]

3.2 CryptDB

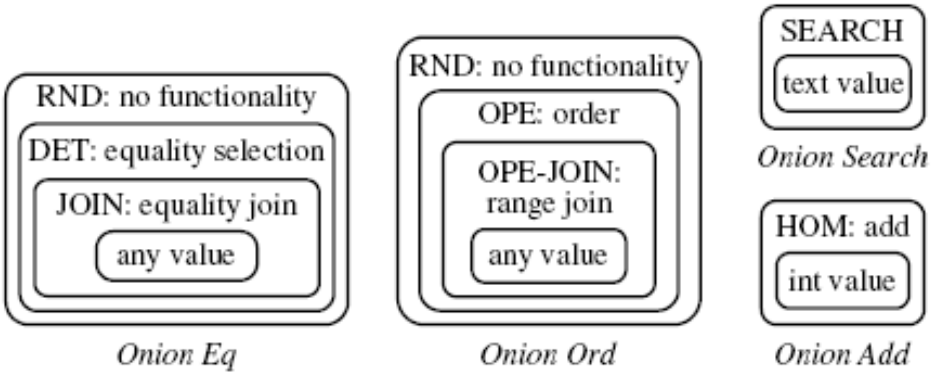
V tejto kapitole sú použité informácie zo zdrojov [56] [48] a [47].

CryptDB je databázový systém, ktorý umožňuje vykonávať SQL dotazy nad zašifrovanou databázou takmer rovnako, ako keby boli vykonávané nad nezašifrovanou databázou. Pri použití CryptDB vystupuje tento systém ako proxy server medzi aplikáciou a databázou. Dáta v databáze sú šifrované. Aplikácia bežiaca u používateľa nad CryptDB môže bežať bez toho, aby sa nejako menila alebo prispôsobovala CryptDB a môže sa správať tak, ako keby bežala na databáze bez šifrovania. Ak používateľ pošle dotaz na databázový server, CryptDB ho zachytí, vykoná na ňom potrebnú transformáciu, resp. šifrovanie a pošle na databázový server. Databázový server pošle zašifrovaný výsledok, CryptDB ho odšifruje a pošle aplikácii používateľa. Vykonanie dotazu teda beží na databázovom serveri.

CryptDB pracuje v rôznych vrstvách. Dôvod je jednoduchý, ak chceme napr. nájsť v databáze ľudí, ktorí majú plat 1000 eur, potrebujeme vedieť robiť porovnávanie reťazcov. Na to nám stačí, ak je celá databáza zašifrovaná deterministicky jedným kľúčom a my si dáme hľadať žiadaný reťazec v zašifrovanej podobe. Ak ale chceme vyhľadať v databáze všetky osoby, ktorých plat je väčší ako 1000 eur, takéto šifrovanie nám stačiť nebude na túto funkcionalitu a treba implementovať aj napr. Order preserving encryption - šifrovanie, ktoré zachováva poradie (t.j. ak $60 < 100$, tak aj enc(60)<enc(100)).

Toto šifrovanie ale odhaľuje viac informácií z databázy. Ak sa šifrujú jednotlivé položky, je dokonca možné uplatniť úplné preberanie a zistiť tak konkrétne hodnoty.

CryptDB má teda viacero vrstiev. Často sú typy prirovnávané k cibuliam a vrstvy nazývané ako vrstvy cibule. Najvrchnejšia vrstva CryptDB odhaľuje najmenej informácií z databázy, ale zato má najmenšiu funkcionalitu, zatiaľ čo najvnútornejšia vrstva poskytuje najväčšiu funkcionalitu za cenu odhalenia niektorých informácií. Prehadzovanie sa medzi vrstvami sa deje na automaticky podľa potreby - napr. keď príde dotaz. CryptDB sa snaží zachovať najvyššiu vrstvu bezpečnosti pokiaľ to je možné. Ak ale príde dotaz, kvôli ktorému treba zvýšiť funkcionalitu, prešifruje požadovaný stĺpec databázy tak, aby bolo na ňom možné vykonať požadovaný dotaz. Následne je technicky možné aj prešifrovať stĺpec naspäť, pre zvýšenie bezpečnosti, toto ale tvorcovia systému neodporúčajú, pretože by to vyžadovalo príliš veľkú výpočtovú silu a okrem toho, tá informácia už bola prezradená, takže nemá veľmi zmysel to opäť prešifrovávať na vyššiu úroveň bezpečnosti.



Obr. 3.1: Vrstvy v CryptDB. (Zdroj obrázku: [48])

Z obr. 3.1 vidíme rôzne vrstvy. Každý typ používa iný algoritmus, ktorý splní dané špeciálne požiadavky konkrétneho typu a každý algoritmus môže byť zvlášť nahradený v prípade potreby, ak napr. niektorá šifra bola prelomená. V tomto prípade treba dáta v daných stĺpcoch odšifrovať starým algoritmom a zašifrovať novým.

Vrstvy z obr. 3.1:

- **RND - Random.** Táto vrstva má najvyššiu bezpečnosť, neobsahuje žiadnu funkcionalitu pre dotazy. Sú v nej dôverné dáta ako napríklad zdravotné diagnózy, čísla kreditných kariet, súkromné správy - teda veci, ktoré nepotrebujú byť porovnávané hodnotami medzi sebou. RND vrstva používa AES šifrovanie (bližšie popísané v kap. 4.4.1) a Blowfish pre šifrovanie čisiel.

- **HOM - Homomorphic encryption.** Táto vrstva sa považuje za rovnako bezpečnú. Povoľuje operácie sčítania nad riadkami daného stĺpca, a teda umožňuje vykonávanie operácií ako sú napr. SUM alebo AVG (ktoré spočítajú súčet, resp. priemernú

hodnotu z hodnôt v stĺpci). Jedná sa teda len o čiastočne homomorfickú schému, ktorá podporuje len operáciu sčítania, nakoľko čiastočné homomorfické šifrovanie je niekoľkonásobne rýchlejšie ako konštrukcie s plne homomorfickým šifrovaním. Používa sa Paillierov čiastočne homomorfický kryptosystém [45].

- **SEARCH - hľadanie slov.** Považuje sa za skoro tak bezpečnú ako vrstva RND. Databáza sa pri hľadaní dozvie koľkokrát (resp. v koľkých riadkoch) sa v nej nachádza hľadané slovo. Podporuje MySQL operácie podobné operácii LIKE. Texty v databáze sa rozdelia na kľúčové slová. Zmažú sa duplikáty a slová sa náhodne premiešajú, aby databáza nevedela zistiť ich pozíciu. Každé slovo sa potom zašifruje s paddingom tak, aby boli všetky slová rovnako dlhé. Používateľ potom pošle dotaz typu: $SELECT*FROMmessagesWHEREmsgLIKE"\%alice\%"$, cryptDB spraví z hľadaného výrazu token, ktorý pošle na server. Server zistí či sa nejaký text zhoduje s tokenom. Takto vieme vyhľadávať iba prítomnosť celého slova, SEARCH nepodporuje ľubovoľné regulárne výrazy.

- **DET - Deterministic.** Vrstva DET je deterministická, a teda rovnaké texty v nej sa zašifrujú na rovnaký šifrový text. Toto nám dovolí používať funkcie ako GROUP BY alebo DISTINCT, ale len na výber riadkov, ktoré nie sú rovnaké. Nepovie nám či sú hodnoty v poliach väčšie alebo menšie. Ako šifrovanie používa opäť AES pre texty väčšie ako 64 bitov, pre kratšie texty používa Blowfish.

- **OPE - Order-preserving encryption.** Vrstava používajúca poradie-zachovávajúce šifrovanie. Na tejto vrstve sú umožnené operácie ako MIN, MAX či ORDER BY. Táto vrstva odhaľuje najviac informácií o dátach. Okrem poradia dát odhaľuje aj blízkosť dát medzi sebou. Toto môže byť akceptovateľné pre niektoré hodnoty, ale pre iné to môže byť problém.

- **JOIN, OPE-JOIN - Join.** Obe tieto vrstvy sú najspodnejšie a poskytujú rôzne výpočtové funkcionality. Vďaka nim môžme vykonať dotaz ako napr. $SELECT*FROMtest_tableWHEREname1 = name2ANDname2 = name3$, kde vykonali porovnanie viacerých hodnôt z databázy (a nie len porovnanie 1 políčka so zadaným reťazcom), pričom OPE-JOIN vie okrem takéhoto porovnania aj povedať, ktorá z hodnôt je väčšia. Ukážkový dotaz však prezradil, ktoré 3 „name“ sú rovnaké.

3.2.1 Nevýhody

Táto konštrukcia je navrhnutá pre veľmi špecifické systémy, kde vieme, že napríklad citlivejšie údaje nebudeme chcieť medzi sebou porovnávať a kde upustiť z bezpečnosti niektorých stĺpcov s hodnotami nám nerobí problém, pretože v nich neboli až tak veľmi

„tajné“ údaje. Taktiež sa odporúča používať vrstvy odhaľujúce poradie dát len pre také stĺpce, kde ich hodnoty majú dostatočne veľkú entropiu. [14]

Konštrukcia odhaľuje rôzne typy informácií na rôznych vrstvách modela serveru a teda aj používateľom a útočníkom. [53]

Taktiež je veľmi potrebné dbať na zabezpečenie proxy servera s CryptDB. Ak by sa podarilo útočníkom kompromitovať tento proxy, tak by mal prístup ku všetkým dátam, ku ktorým môžu pristupovať aktuálne prihlásení používatelia a získať ich kľúče. [41] Proxy v tomto modeli vystupuje ako dôveryhodná „tretia strana“ a je to len aplikácia naprogramovaná ľudmi, ktorá môže obashovať rôzne chyby. Nato si tiež treba dávať pozor. [53]

Ďalším problémom môže byť veľkosť. Pri tvorení tabuľky proxy najskôr zistí akého typu sú dáta v nej a podľa toho, aké operácie je na nich možné vykonávať následne vytvorí niekoľko stĺpcov pre každý stĺpec (pre jednotlivé „cibule“ z obr. 3.1). Napríklad tabuľka (3.2) obsahujúca ID a meno bude mať v konečnom dôsledku 8 stĺpcov (3.3). Napr. keďže ID je číslo, nebude mať „cibuľu“ pre SEARCH a keďže meno je string, nebude mať „cibuľu“ pre HOM. [23]

ID	Meno
1	Alice
2	Bob
3	Eve

Tabuľka 3.2: Tabuľka databázy, ktorú chceme vytvoriť v CryptDB, zdroj: [23]

C1-IV	C1-Eq	C1-Ord	C1-Hom	C2-IV	C2-Eq	C2-Ord	C2-Search
q39f	ge88	hwip	dwna	lpw9	dkfm	fdkw	98wu
c8x3	8n4o	s09s	28bd	suc0	d82w	8mx0	x9ak
sk7x	x7wk	x6sh	s7w9	3mnu	snge	xuwn	u8sb

Tabuľka 3.3: Tabuľka databázy, ktorú vytvoril proxy CryptDB, zdroj: [23]

Podobne tabuľka pre študentov obsahujúca ID, meno a počet bodov z testu bude vo výsledku tabuľka s 12 stĺpcami. CryptDB má teda veľký overhead. Taktiež je absolútne nepostačujúca funkcia SEARCH, ktorá nevykonáva ani zďaleka potrebnú funkcionalitu na vyhľadávanie v databáze. [23]

V práci autorov Ishan H. Akin a Berk Sunar [32], ktorí pracovali na skúmaní bezpečnosti CryptDB sú taktiež nie veľmi priaznivé výsledky. Skúmali webové aplikácie

používajúce CryptDB na komunikáciu s cloudom v nastavení pre viacero používateľov. Tvrdia, že útočník alebo správca databázy môžu získať citlivé údaje používateľov cez prístup k jedinému ľubovoľnému používateľskému účtu. Spravili tak zásahom do integrity správ, ktoré sú posielané medzi databázou a proxy serverom počas komunikácie. Čo je prekvapujúce je fakt, že dokážu získať údaje alebo meniť privilégiá bez akéhokoľvek útoku na proxy server alebo webový server s aplikáciou. V týchto útokoch sa vôbec nezaoberali OPE šifrou, ktorú označili za najzraniteľnejší a najslabší článok CryptDB. V práci navrhli riešenia týchto problémov.

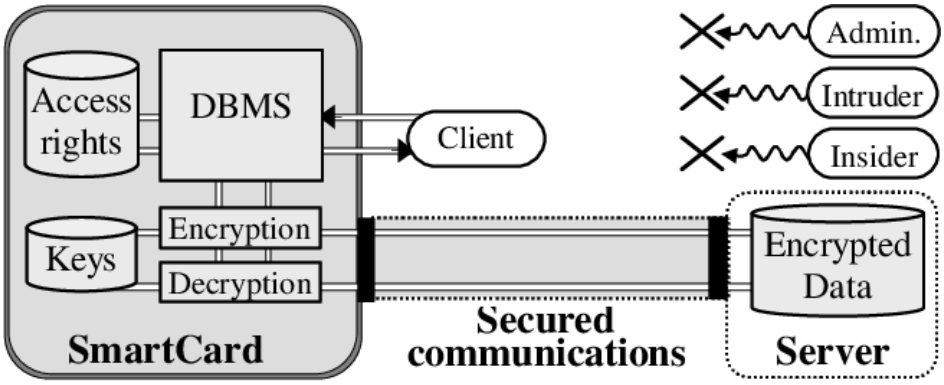
Ďalej v práci uviedli skutočnosť, že útočníci vedia skenovať databázu s cieľom zisku citlivých údajov pomocou neoprávnených zásahov do dotazov od používateľov.

Nakoniec ešte v práci demonštrovali frekvenčný analýzový útok (frequency analysis attack).

3.3 C-SDA

Informácie v tejto kapitole pochádzajú zo zdrojov [9] a [57].

C-SDA alebo tiež Chip secured data access je model založený na myšlienke izolovať šifrovanie, vyhodnocovanie dotazov a prístupové práva od servera a vykonávať ich pomocou špeciálneho hardvéru, obr. 3.2. Týmto hardvérom je „inteligentná karta“.



Obr. 3.2: C-SDA intelingentná karta. (Zdroj obrázku: [9])

Inteligentná karta by mala byť umiestnená pred vstupom do cloudu, teda na strane cloudu, ako dôveryhodný prostredník. Tento model je podobne ako CryptDB založený na rozdelení dotazov. C-SDA rozdľuje dotazy na poddotazy.

- **Poddotazy pre server.** Dotazy, ktoré vedia byť vykonané cloudom bez narušenia dôvernosti dát. Tieto dotazy vedia poskytnúť podobnú funkcionalitu ako DET šifrovacia vrstva v CryptDB, ktorá je založená na deterministckých operáciách. C-SDA tu používa blokové šifrovacie algoritmy ako DES alebo Blowfish. Tieto vedia

zapezbečiť funkciu overenia rovnosti so zadaným reťazcom na zašifrovaných dátach. Toto však odhaľuje istú mieru korelácie medzi dátami. V tejto časti dovoľujú použiť autori aj OPE - šifrovanie zachovávajúce poradie, v prípde ak potrebujeme vykonávať efektívne aj dotazy na porovnávanie hodnôt, avšak znižujú sa opäť záruky dôvernosti dát a prezrádzajú sa ďalšie informácie o dátach.

- **Poddotazy pre kartu.** Karta podporuje dotazy založené na primitívnych operátoroch, ktoré server nepodporuje. Umožňuje homomorfické operácie ako agregáciu dát a tiež funkcie na šifrovanie a dešifrovanie. Toto vytvára výpočtové zaťaženie pre túto inteligentnú kartu. Komplexné dotazy preto nie sú efektívne pri použití pamäte RAM. Toto sa nemôže diať u klienta, karta pracuje s výsledkami od servera, ku ktorým klient nemusí mať prístupové práva.

- **Poddotazy pre klienta.** Klient vyhodnocuje len poddotazy, ktoré sa týkajú „prezentácie výsledku“. To je napríklad triedenie (sort) alebo odstraňovanie duplikátov a filtrovanie.

Výkonnosť záleží na inteligentnej karte, ktorá má obmedzenú výpočtovú silu a pamäť.

V dnešných dňoch však majú tieto inteligentné karty slabú výkonnosť v oblasti kryptografie. Bežné typy inteligentných kariet súčasnosti majú slabý výkon pre operácie ako hašovanie, generovanie náhodných čísel, modulárne aritmetické operácie na Big integer-och, ktoré zlyhávajú v dosahovaní rozumných časov vykonania. Poukázali na to benchmarky vedcov z univerzity v Brne [30].

Inteligentné karty sa dajú kúpiť za zanedbateľnú čiastku, pohybujúcu sa okolo pár dolárov. Problémom však môže byť potreba umiestnenia tohto hardvéru do blízkosti servera. Karty majú dostatočne zabezpečenú ochranu pred fyzickými útokmi. Avšak boli navrhnuté rôzne útoky analýzou dát - Simple Power Attack (SPA) alebo Differential power analysis (DPA).

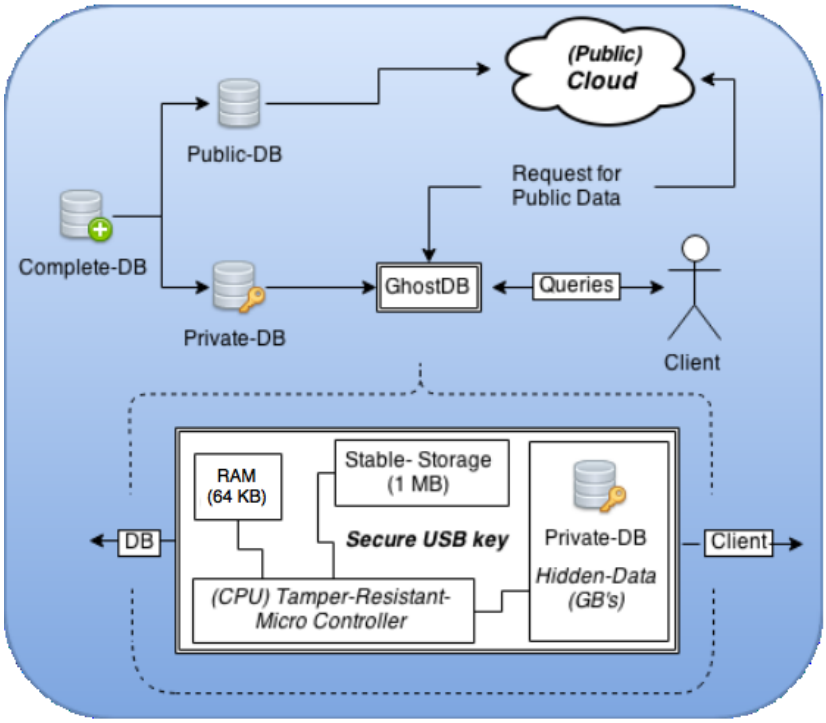
3.4 GhostDB

V tejto kapitole sú informácie zo zdrojov [5] a [57].

GhostDB je schéma, ktorá sa spolieha na hardvér pre zaistenie dôvernosti dát, podobne ako schéma C-SDA. Pri GhostDB je však tento hradvér na strane klienta a nepotrebuje žiadnu bonusovú kooperáciu so strany servera.

GhostDB rozdelí databázu na 2 časti, verejnú a súkromnú. Len verejná časť je uložená na „nedôveryhodnom“ cloude, zatiaľ čo súkromná časť databázy je uložená v hardvéri GhostDB. V prvotnej verzii predstavoval tento hardvér USB 2.0 kľúč, ktorý

obsahoval 1 MB stabilného úložného priestoru, 64 KB RAM, CPU a pár GB priestoru pre súkromnú časť databázy, obr. 3.3.



Obr. 3.3: GhostDB architektura. (Zdroj obrázku: [57])

Klient posiela dotazy úplne rovnako ako na nezabezpečenú databázu a GhostDB spracuje tento dotaz. Ak v dotaze nevystupujú žiadne dáta z verejnej časti databázy, nekomunikuje so serverom, dotaz vykoná len lokálne u seba. Ak boli v dotaze žiadané dáta, ktoré sa nachádzajú v súkromnej časti databázy pristupuje do tejto časti databázy uloženej u seba. Použité techniky ako climbing indexes, subtree key tables a Bloomové filtre (tiež popísané v [5]) zlepšujú výkonnosť tohto systému a dovoľujú vykonávať aj náročnejšie dotazy nad rozsiahlejšími databázami. GhostDB nemá žiadne obmedzenia na dotazy a vie vykonať všetky SQL operácie, ktoré dokáže vykonať aj počítač alebo server. Umiestnenie časti dát na určitom nosiči však obmedzuje využitie výhod cloudu.

GhostDB je možné použiť aj v nezabezpečenom prostredí, ako napr. z verejného, nedôveryhodného počítača len s hrozbou úniku výsledku vyžiadaného dotazu.

Ďalším obmedzením je samotný nosič. Tento je limitovaný vlastnou veľkosťou pamäte RAM a výpočtovou silou svojho CPU, čo môže ovplyvniť výkon a rýchlosť vykonávania dotazov.

Takisto nastáva problém pri použití viacerými používateľmi, keď by chceli vykonávať dotazy, ktoré menia obsah databázy.

Veľké nebezpečensvto taktiež nastáva pri strate alebo odcudzení nosiča, pretože práve na ňom je uložená celá súkromná časť databázy.

3.5 Helib

Helib je softvérová open-source knižnica, ktorú začal vyvíjať Shai Halevi v roku 2013. Používa homomorfické šifrovanie. Je napísaná v C++ a používa NTL knižnicu. Snaží sa implementovať verziu homomorfického šifrovania, ktorá je založená na RLWE - ring learning with error - navrhnutého autormi Brakerski, Gentry a Vaikuntanathan (BGV) [12]. O algortimoch a detailoch v tejto knižnici sa môžete dočítať v práci autorov Shai Halevi a Victor Shoup [31].

Helib bola prvou takouto knižnicou, dnes už existuje minimálne 10 open-sourcových knižníc, ktoré sa pokúšajú implementovať homomorfické šifrovanie. Porovnanie niektorých z nich si môžte prečítať v práci autora Sai Sri Sathya a spoluautorov [54].

Keďže sa tieto knižnice snažia implementovať homomorfické šifrovanie, zdedili z neho niektoré nevýhody, ktoré sa snažia rôznym spôsobom riešiť. Používajú rôzne metódy na riešenie problémov pri rôznych verziách a prístupoch k homomorfickému šifrovaniu. Napriek rýchlemu napredovaniu v oblasti homomorfického šifrovania v teoretickej rovine, je potrebné robiť pokroky aj v praktickej forme jeho použitia. Vo veľa prípadoch bránia v praktickom zavedení homomorfického šifrovania práve technické a právne dôvody. [54]

Kapitola 4

Návrh riešenia

V tejto kapitole naznačíme a navrhneme riešenie, ktoré zabezpečí ochranu osobných či iných údajov v cloude. Pri tom budeme dbať na to, aby riešenie dovoľovalo prístup viacerých používateľov, a teda sa dalo použiť vo firme s viacerými zamestnancami, ktorí s údajmi pracujú. V prvom rade však dbáme na to, aby údaje boli dostatočne chránené podľa zákonov SR, zabezpečíme, aby sa poskytovateľ cloudu nikdy nedozvedel heslo na odšifrovanie údajov a aby sme vedeli vykonávať operácie, ktoré musíme vedieť zo zákona vedieť vykonávať pri spracovávaní osobných údajov (kap.2). Taktiež navrhneme aj spôsob, ktorý následne zefektívni prácu s týmito údajmi.

Porovnanie našich riešení s existujúcimi riešeniami z kap. 3 je stručne popísané závere.

Pri každom uvedenom návrhu najskôr pojednáme, ktoré požiadavky spĺňa, popíšeme jeho vnútorné vlastnosti a následne na príklade ukážeme ako by mohla prebiehať komunikácia používateľa s cloudom.

Návrhy sa môžu implementovať ako samostatná aplikácia, ktorú si používateľ musí nainštalovať, prípadne sa môžu implementovať s využitím internetového prehliadača, keďže cloudové služby sú prístupne aj cez web.

4.1 Úvod a motivácia

Snažíme sa navrhnúť riešenie, pri ktorom bezpečnosť osobných údajov alebo iných tajných informácií bude v prvom rade čo najmenej závislá na použitom softvéri. Vieme, že je takmer nemožné naprogramovať rozsiahlejší softvér bez rôznych nedostatkov, chýb a bugov. [69]

Taktiež je veľmi obľúbené vytváranie backdoorov v aplikáciách, t.j. „zadných vrát“. Tieto backdoory slúžia na to, aby mali tvorcovia takejto aplikácie pri použití nejakým človekom istú výhodu alebo možnosť kontroly a aby sa vedeli dostať do systému, kde je softvér nainštalovaný a používaný. Backdoor môže byť v aplikácii nevinne, len ako kra-

jná možnosť, v prípade potreby verejnej moci alebo na sledovanie správania aplikácie a prípadnej opravy alebo sledovanie správania používateľa na vylepšenie aplikácie alebo na reklamné účely či na preniknutie do jeho systému a spustení ľubovoľného softvéru. Dokonca aj známa a veľká spoločnosť Apple tvrdí, že FBI sa ich snažila prinútiť zabudovať do svojich produktov backdoory [52]. Backdoor bol taktiež aj v IBM aplikácii Lotus-Notes (ktorá slúži ako emailová služba, plánovač rozvrhov, na zdielanie dokumentov, atď.), ktorý bol backdoorovaný Národnou bezpečnostnou agentúrou - NSA. Časť kľúča RSA šifrovania v tejto aplikácii bola tajne pevne určená spoločnosťou NSA, a tá sa potom mohla ľahko dopočítaním zvyšku kľúča dostať k celému kľúču a následne obsahu [2]. Tieto backdoory však v prípade odhalenia môžu byť využité aj útočníkom alebo hackerom. Viac o backdooroch si môžete prečítať v zdrojoch [42] [70].

Aj v prípade, že by bola napr. konkrétna aplikácia spravujúca a šifrujúca naše údaje v cloude relatívne poctivo a správne naprogramovaná, môžu byť rôzne diery a slabiny v zdrojoch, ktoré využíva, ako napr. knižnice, ktoré tiež nie sú vždy úplne bez chýb či iné aplikácie, ktoré treba na jej beh, ako napr. operačný systém, ktorý vieme tiež, že obsahuje veľa bugov a chýb a býva pravidelne zaplatávaný v aktualizáciách.

Chceme navrhnúť riešenie, ktoré bude závislé len na sile šifrovania a na šifrovaní samotnom a následná potrebná manipulácia s dátami z cloudu bude prebiehať lokálne napríklad aj offline a odrezaná od zvyšku sveta. V tomto príapde sa teda cloud nesmie nikdy dozvedieť a spravovať heslo k našim dátam, pretože by mohlo byť či už útočníkom alebo zamestnancom cloudu získané pomocou možných bugov a backdoorov cloudového softvéru a iných súčastí servera.

Ďalšou výhodou tejto požiadavky je, že ak by boli údaje spravované a spracovávané cloudom, tak by sme si museli dávať pozor na výber cloudovej spoločnosti a to, v ktorej krajine konkrétne má umiestnené svoje servery, prípadne na to, či ich cloudová spoločnosť niekedy počas nášho využívania nepresunie do nejakej inej tretej krajiny.

Lokálnym šifrovaním môžme utlmiť aj dopady únikov dát, tzv. data breaches. Úniky dát spoločností nastávajú niekoľkokrát ročne a nevyhli sa im ani veľké spoločnosti uchovávajúce osobné údaje týkajúce sa niekoľko miliónov používateľov. Najčastejším dôvodom úniku dát býva napadnutie databázy s údajmi hackerom, ďalšími dôvodmi sú: slabá bezpečnosť, strata či ukradnutie fyzického média s dátami či počítača, náhodná publikácia dát na verejnej stránke, nepoctivý zamestnanec firmy, zlá konfigurácia databázy a podobne. Ako príklad môžeme uviesť firmu eBay, ktorej v máji 2014 útočníci hackli a zverejnili databázu obsahujúcu 145 miliónov používateľských mien, adries, dátumov narodení a zašifrovaných hesiel či firmu Yahoo a s ňou spojený najväčší únik dát v histórii. V septembri 2016 firma Yahoo zistila, že v roku 2014 útočníci kompromitovali reálne mená, emailové adresy, dátumy narodení a telefónne čísla 500 miliónom používateľov. Následne na to v decembri roku 2016 zistili iný únik dát z roku 2013 inými útočníkmi, ktorý kompromitovali mená, dátumy narodení, emailové adresy, heslá a tak-

tiež aj bezpečnostné otázky a odpovede k nim 1 miliardy používateľských účtov. Tento údaj v októbri 2017 opravili na 3 miliardy kompromitovaných účtov [67] [72]. Chronologicky zoradené úniky dát s rôznymi filtrami a odkazmi si môžte pozrieť v [22]. Tomuto v našom prípade zabránime, pretože budú všetky osobné údaje v databáze šifrované.

4.2 Požiadavky a ich naplnenie

Požiadavky na model, ktorý by mal zabezpečiť ochranu osobných údajov v cloude sme rozdelili do dvoch kategórií - bezpodmienečné a výberové. Bezpodmienečné sú také, ktoré sú dané zo zákona a/alebo je žiadané, aby ich náš model spĺňal. Výberové sú také, ktoré vylepšujú a dopĺňajú ďalšie možné požiadavky na model, aby boli prakticky použiteľné vo väčších a rôznorodých inštitúciách a firmách.

- Bezpodmienečné:
  - dôvernosť dát,
  - integrita a autentickosť,
  - možnosť jednoducho a efektívne meniť a opravovať údaje
  - efektívne vyhľadávanie všetkých údajov týkajúcich sa jednej osoby
  - prístup viacerých používateľov k rôznym častiam dát
  - dostupnosť a odolnosť systémov s údajmi
  - cloud nemá prístup k šifrovacím kľúčom
  - logovanie
- Výberové:
  - podpora skupín, skupiny používateľov majúcich možnosť pristupovať k tej istej skupine údajom
  - rozdielne práva rôznych pristupujúcich - (len čítanie, aj písanie,..)

4.2.1 Bezpodmienečné

Dôvernosť

Dôvernosť údajov znamená, že údaje sú prečítateľné len tými osobami, ktoré na to majú právo a neoprávnené osoby tieto údaje (a informácie obsiahnuté v nich) nevedia získať v čitateľnej forme. Túto požiadavku zabezpečíme šifrovaním všetkých údajov v databáze (toto priblížime v jednotlivých návrhoch). Neodšifrované ostanú len ID čísla záznamov, vďaka ktorým si používateľ môže pýtať záznamy.

K rôznym častiam databázy môže pristupovať viacero rôznych ľudí, ktorí ale nemajú práva vidieť celú databázu, napr. je viacero pracovníkov na finančnom oddelení firmy a každý má prístup len k osobným údajom skupiny ľudí, ktorú on spravuje. Prípadne ak máme napríklad databázu študentov (ale aj iných osôb), tak chceme, aby okrem prístupu učiteľov a študijných referentiek k skupinám študentov, vedeli pristupovať k svojim údajom aj samotní študenti, a tak si napríklad overiť ich korektnosť a zároveň sa nevedeli dostať k osobným údajom iných študentov. Preto a aj pre ďalšie potreby zašifrujeme každý záznam v databáze iným kľúčom. Tieto kľúče sú generované na začiatku administrátorom systému, nie cloudom (bližšie popísané ďalej v tejto kapitole).

Integrita a autentickosť

Zaručením integrity sa rozumie zaručiť, že údaje, ktoré sú v databáze sú tie, ktoré tam boli pôvodne dané a neboli nijako zmenené či už úmyselne - neoprávnenou osobou alebo náhodne. Autentickosťou zaručujeme to, že údaje pochádzajú od skutočného autora a od vytvorenia neboli nijako poškodené.

Toto môžme zabezpečiť v komplikovanejšom prípade vyžitím asymetrického šifrovania a digitálneho podpisu pre každý záznam v databáze. V jednoduchšom prípade, ak chceme používať len jeden kľúč, postačí ako riešenie autentizačný tag, akým je napríklad HMAC (bližšie popísaný v kap. 4.4.2). Tento na základe textu a kľúča vytvorí hash. Tu ale pozor, pri niektorých šifrách nesmieme použiť ten istý kľúč viackrát, použijeme teda buď iný kľúč (iný od toho, ktorý bol použitý na zašifrovanie dát) alebo pôvodný sa bude skladať z dvoch častí - jedna časť pre záznam a druhá časť pre autentizačný tag/hash. Takýto hash zabezpečí aj autentickosť, pretože ho vedel vytvoriť len človek, ktorý poznal kľúč, a teda mal na to oprávnenie. Ak však vlastní kľúč viacero ľudí manipulujúcich s databázou, nevieme určiť, ktorý konkrétne to bol, len, že to nebol nikto neoprávnený.

Efektívna manipulácia s údajmi

Pri tomto bode nám hlavne záleží na efektívnosti akcií ako vymazanie či úprava údajov, ktoré dotknutým osobám musíme vedieť zo zákona poskytnúť. Pri týchto operáciách sa však nevyžaduje nič špeciálne, pri úprave údajov sa tieto údaje len zašifrujú kľúčom, ktorý upravovateľ má a spraví sa z dát digitálny podpis alebo autentizačný tag a takto sa pošlú nové dáta naspäť na cloud. Nemusí sa prešifrovávať celá databáza ani žiadne kľúče, upravovateľ zašifruje len konkrétny menený záznam.

Efektívne vyhľadávania

Efektívnosť pri narábaní je zabezpečená v prvom rade indexovaním každého záznamu. Keďže všetky údaje sú šifrované, vyhľadávanie podľa plaintextových hodnôt a reťazcov

v nich je komplikovanejšie. O to viac, keď máme každý záznam zašifrovaný iným kľúčom K. Zvýšeniu efektívnosti sa viac venujeme v kap. 4.7.

Prístup viacerých používateľov k určeným častiam dát

Každý záznam budeme teda šifrovať iným kľúčom K. Zoberme si ako príklad študentov univerzity a ich prístup k vlastným údajom, ktoré o nich univerzita má. Kľúčom na zašifrovanie údajov študenta by teda mohlo byť heslo tejto dotknutej osoby, avšak chceme, aby k údajom vedeli pristupovať následne aj iné osoby alebo skupiny osôb, ktoré s týmito údajmi potrebujú manipulovať a používať ich, preto šifrovanie údajov heslom dotknutej osoby neprichádza do úvahy. Každý záznam teda zašifrujeme nejakým náhodným K. Tieto K potom uložíme do príslušnej tabuľky v zašifrovanej podobe a zašifrované heslom toho, kto k nemu chce pristupovať a má na to právo. Tabuľka, nazvyme ju „Tabuľka prístupu“, teda bude vyzerať takto 4.1.

Accessor ID	Accessed ID	encrypted K
(Id toho kto pristupuje)	(Id toho k čomu pristupuje)	(K zašifrované heslom toho, kto pristupuje)

Tabuľka 4.1: Tabuľka prístupu

Ak by teraz chcela nejaká osoba pristúpiť k údajom, vypýta si z tejto tabuľky prístupov požadovaný riadok - svoje ID uvedie ako „Accessor ID“, ID toho, k čomu chce pristúpiť uvedie ako „Accessed ID“. Server zašle tento riadok, používateľ si riadok odšifruje svojim heslom u seba a tak získa kľúč K. Následne požiada server o (resp. už ho žiadať nemusí a server mu automaticky pošle na základe prijatého „Accessed ID“) údaje k množine „Accessed ID“, teda dáta s ID číslom, ku ktorým chcel pristupovať. Používateľ dostane požadované dáta v zašifrovanej forme a tieto si u seba odšifruje s už získaným K.

Dostupnosť a odolnosť

Dostupnosť a odolnosť je vyriešená výberom cloudu. Cloudy v súčasnosti poskytujú pevné a dostatočné záruky dostupnosti aj odolnosti.

Cloud nepozná kľúč

Cloud sa v priebehu spracúvania osobných či tajných údajov nikdy nedozvie kľúč na odšifrovanie dát ani obsah dát uložených na ňom. Toto sme dosiahli tak, že údaje sa na cloud posielajú už v zašifrovanej forme, cloud ich má celý čas zašifrované, na

požiadanie používateľa mu pošle dáta, ktoré má uložené u seba s požadovaným ID v zašifrovanej forme. Používateľ si odšifruje prijatý záznam až lokálne a s kľúčom, ktorým boli pôvodne zašifrované.

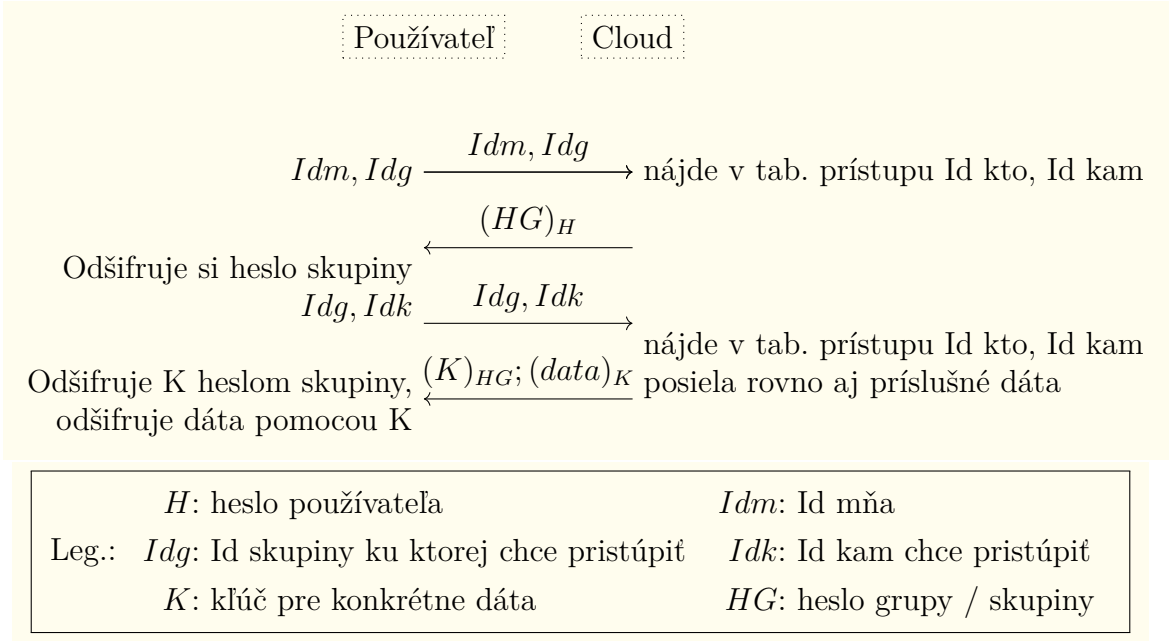
Logovanie

Logovanie, ktoré je v prípade verejného orgánu alebo verejnoprávnej inštitúcie podľa zákona uvedeného v kapitole 2.2.7 povinné, môžme zabezpečiť na strane cloudu. Tento bude v logoch zaznamenávať kto si kedy pýtal ktoré konkrétne dáta z databázy. Ako informáciu „kto“ si môže zaznamenávať cloud podľa možnosti ip adresu, mac adresu pristupujúceho zariadenia, prípadne stačí len meno/ID potencionálneho pristupujúceho také, akým sa snažil pristúpiť k svojim údajom, a teda aké si pýtal z cloudu ako „Accessor ID“, kedže každý človek si musí najskôr vypýtať riadok konkrétne pre seba ako jednotlivca s ďalšími zašifrovanými kľúčami K.

4.2.2 Výberové

Skupiny

Tu uvedieme malé vylepšenie efektívnosti. Aby nemuselo mať veľa používateľov cloudu veľa riadkov v tabuľke prístupu podstate k tým istým riadkom databázy, môžme vytvoriť skupiny používateľov, ktoré môžu pristupovať k rôznym dátam. Uvedieme na príklade o škole - na predmet je zapísaných 100 študentov, učiteľ má prístup k ich údajom, aby im vedel zapísať známky. Na predmet boli pridelení ďalší dvaja učitelia/cvičiaci, ktorí majú tiež prístup k presne tej istej množine 100 študentov - do tabuľky prístupu pribudne ďalších 2x100 riadkov? Nemusí. Vytvoríme skupinu, označme ju G1. Teraz len skupina G1 má prístup a záznam ku všetkým 100 študentom v tabuľke prístupov. Skupina G1 má vygenerované vlastné náhodné heslo, ktorým si kľúč K vie odšifrovať. Učitelia/cvičiaci predmetu teraz každý z nich má len jeden riadok v tabuľke. Ako Accessed ID, teda to, k čomu pristupuje, je uvedená skupina G1. V políčku s kľúčom K je teraz kľúč skupiny G1 zašifrovaný učiteľovým heslom. Učiteľ si vypýta tento riadok, odšifruje si u seba kľúč/heslo skupiny G1 a teraz si vypýta riadok s Accessor ID G1 k tomu, k čomu chce pristupovať. Databáza mu pošle zašifrovaný riadok z tabuľky prístupu a zároveň dáta s požadovaným Accessed ID. Učiteľ si získaným heslom skupiny odšifruje kľúč K k dátam. V tomto príklade sme potrebovali namiesto 300 riadkov len 103, avšak bol potrebný jeden prístup do tabuľky prístupov navyše, čo by však nemal byť problém, pretože táto tabuľka je utriedená podľa ID čísiel. (Obr. 4.1)

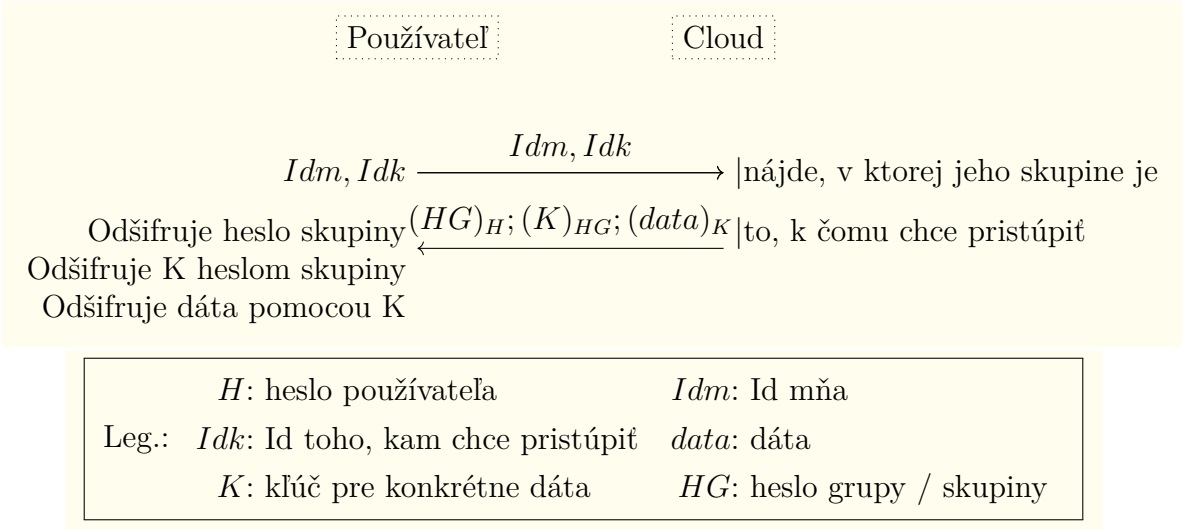


Obr. 4.1: Podpora skupín v databáze.

Ďalšou možnosťou ako spravovať skupiny cloudom, je mať uloženú tabuľku používateľov a ich skupín samostatne. (Tab. 4.2) Toto samostatné rozdelenie nijako neznižuje bezpečnosť, id čísla boli rovnako nezašifrované aj v ostatných tabuľkách. Užívateľ si tak nemusí pamätať v akých skupinách je a posiela len svoje id číslo a id číslo toho, k čomu chce pristupovať. Cloud si následne sám zistí, v ktorých skupinách je používateľ a akým spôsobom môže pristúpiť k tomu, čo požaduje a poslať mu tak potrebné dáta. (Obr. 4.2)

Kam chcem pristúpiť	Kto má prístup
Id skupiny/dát	Id používateľa/skupiny

Tabuľka 4.2: Tabuľka priradenia skupín.



Obr. 4.2: Podpora skupín v databáze verzia 2.

Práva - len čítanie a čítanie aj písanie

Ďalším problémom je, že niektorí používatelia alebo zamestnanci môžu tabuľku len vidieť a čítať a iní môžu do tabuľky aj zapisovať. Tu si ukážeme len jednoduché názorné riešenie ako by sa to mohlo riešiť.

Môžme vyskúšať namiesto jedného K použiť asymetrické šifrovanie s 2 kľúčmi K - jedným verejným a jedným súkromným. Verejným kľúčom sa dáta šifrujú. Osoby, ktoré môžu dáta len čítať budú mať prístup len k súkromnému kľúču. Osoby, ktoré vedia aj zapisovať, budú mať prístup k obom kľúčom. Tu nastáva problém overenia správnosti údajov, teda overenie integrity a autentickosti. Tieto môžeme vyriešiť hashom alebo digitálnym podpisom. Kľúč na vytvorenie digitálneho podpisu bude uvedený ako súčasť alebo časť verejného kľúča. Tabuľka prístupu by sa teda rozšírila 4.3.

Accessor ID	Accessed ID	encrypted private K	encrypted public K
(Id toho kto pristupuje)	(Id toho k čomu pristupuje)	(súkromné K zašifrované heslom toho, kto pristupuje)	(verejné K zašifrované heslom toho, kto pristupuje)

Tabuľka 4.3: Tabuľka prístupu aj s právami rw, verzia 1

Tu ale môže byť problém, že pri niektorých asymetrických šifrách a nastaveniach je možné vypočítať verejný kľúč zo súkromného. Asymetrické šifrovanie je vo všeobecnosti taktiež niekoľko-násobne pomalšie ako symetrické. Preto navrhujeme aj iné riešenie práv na zápis. Použijeme na to len digitálne podpisy. Používatelia, ktorí môžu meniť

príslušný záznam, budú mať v tabuľke prístupu aj kľúč na vytvorenie digitálneho podpisu. Ostatní používatelia, ktorí majú len práva na čítanie si budú môcť digitálny podpis len overiť.

O digitálnych podpisoch si povieme viac v kap. 4.6.1. V praxi to znamená toľko, že budú pre digitálny podpis 2 kľúče, jeden súkromný - tým používateľ vytvára digitálny podpis k dokumentu a zaručuje tak ním integritu a autentickosť dokumentu a druhý verejný, ktorý slúži na overenie toho, že podpis bol vytvorený korektne a so záznamom nikto iný nemanipuloval. Verejná časť kľúča na overenie digitálneho podpisu môže byť v tabuľke prístupu zreťazená z kľúčom K na odšifrovanie dát. Súkromný kľúč K_D na vytvorenie digitálneho podpisu bude umiestnený v ďalšom stĺpci. Používatelia, ktorí nemajú právo na zápis v tomto zázname, budú mať v stĺpci so súkromným kľúčom hodnotu NULL.

Accessor ID	Accessed ID	encrypted $K\|\|K_{vk}$	encrypted K_D
(Id toho kto pristupuje)	(Id toho k čomu pristupuje)	($K\|\|K_{vk}$ zašifrované heslom toho, kto pristupuje, K_{vk} je kľúč na overenie digitálneho podpisu)	(K k vytvoreniu digitálneho podpisu, zašifrované heslom toho, kto pristupuje, prípadne hodnota NULL)

Tabuľka 4.4: Tabuľka prístupu aj s právami rw, verzia 2

4.3 Ďalšie poznámky a problémy

V prvom rade by sme chceli zdôrazniť dôležitosť výberu, použitia a implementácie šifrovania a šifrovacích funkcií, keďže bezpečnosť našich údajov je závislá hlavne na ich sile a korektnosti. Nemôžme zanedbať túto skutočnosť, a preto vybrať zodpovedne overenú a silnú šifrovaciu funkciu, takisto ako aj generátor pseudonáhodných čísel pri generovaní náhodných kľúčov K k záznamom, napríklad z už schválených a štandardizovaných takýchto funkcií NIST-om. V žiadnom prípade neodporúčame takéto funkcie implementovať vlastnoručne alebo zasahovať do kódu už overených takýchto funkcií, čo by mohlo mať pre bezpečnosť údajov fatálne následky, ako sme na to experimentálne poukázali v našej bakalárskej práci „Dôsledky zmien v kryptografických algoritmoch“ [65].

4.3.1 Ukážka vzhľadu databázy

Tabuľka s osobnými údajmi alebo dátami v cloude bude vyzerať v návrhoch riešení nasledovne - tab. 4.5.

ID	Dáta... (zašifrované)	Autentizačný tag/digitálny podpis
Id tohto záznamu/študenta	Potrebný počet stĺpcov alebo serializované dáta...	Autentizačný tag prípadne digitálny podpis

Tabuľka 4.5: Názorná tabuľka s dátami

Naše dáta v nej nemusia byť uložené v jednotlivých stĺpcoch, môžu byť aj serializované, napr. vo formáte JSON a takto zašifrované. Používateľ si ich po odšifrovaní jednoducho deserializuje naspäť do formátu tabuľky.

4.3.2 Problém vlastnenia získaného kľúča

Problémom, ktorý nás môže trápiť, môže byť skutočnosť, že ak si nejaký zamestnanec firmy vypýta z cloudu dáta, ku ktorým ma prístup a dostane k nim kľúče K tak sa dozvie tieto K v podstate nastálo. Keď takýto zamestnanec z firmy odíde alebo ho vyhodia, kľúče od dát firmy by si mohol zapísať a stále ich poznať. To znamená, že by mohol aj naďalej nielen čítať ale aj prepisovať dáta a vytvárať k nim hashe či digitálne podpisy. Štandardne by mu to ale cloud umožniť nemal, takýto zamestnanec stratí prístup do cloudu, avšak môžu vzniknúť rôzne problémy napr. zamestnávateľ mu prístup neznemožnil hneď alebo tieto zmeny v cloude ešte neaktualizoval, zamestnanec bol už prepustený, ale je ešte vo výpovednej dobe alebo v systéme je nejaká zraniteľnosť, prípadne zamestnanec je stále zamestnancom, len mal stratiť prístup k nejakej skupine údajov a už spravuje inú skupinu údajov. Pri každej takomto odchode zamestnanca alebo preskupeniu používateľa na práva k iným údajom by preto bolo potrebné prešifrovať všetky dáta, ku ktorým mal prístup v tabuľke dát a následne aj v tabuľke prístupu, a to všetkým používateľom, prípadne skupinám, ktoré mali tiež prístup k týmto dátam a aktualizovať všetky príslušné uložené kľúče. Toto prešifrovanie by zase muselo prebiehať lokálne administrátorom.

Druhou možnosťou by bolo zabezpečiť, aby sa používateľovi, ktorý už nemá mať prístup k istým údajom (a predtým mal), tieto údaje viac už ani neposlali. To môžme zabezpečiť prihlasovaním sa a overovaním totožnosti. Ale ako sme už spomínali, nechceme sa spoliehať hlavne na softvér, ktorý by mohol obsahovať diery a zraniteľnosti.

Taktiež môžme zamedziť posielaniu len konkrétnych údajov z tabuľky s dátami. Tieto budú poslané len ak používateľ uvedie svoje ID a cloud zistí, že človek s daným

ID má k záznamom naozaj prístup či už priamo alebo cez nejakú skupinu. Takto by „zlomyseľný vyhodený zamestnanec so zapamätaným K“ musel hádať ID číslo, ktoré má k záznamu prístup. Cloud tabuľku prístupu neposiela nikomu na požiadanie. Niekoľko neúspešných pokusov o prístup k tým istým dátam s rôznym „Accessor ID“ je netypické správanie, ktoré je zapísané aj v logoch a na ktoré môže byť upozornené.

Toto môžme vyriešiť aj tak, že kľúč K na odšifrovanie dát sa k používateľovi v otvorenej forme nikdy nedostane a využíva ho len ako keby „nepriamo“. (Viac v návrhu 2 v kapitole 4.5). Toto riešenie je však veľmi špecifické a špeciálne navrhnuté pre riešenie tohto problému.

4.3.3 Problém správy kľúčov a ich obnova

Nie je nezvyčajné, že ľudia z času na čas zabudnú svoje vytvorené heslo na prístup do nejakého systému. Toto spôsobí v našom návrhu problém, nakoľko niektoré záznamy v cloude môžu byť zašifrované práve týmto jediným heslom a nikto iný sa k nim nevie dostať. Strata takýchto dát by bola nezvratná.

Záloha používateľských hesiel v cloude neprichádza do úvahy, práve tomu sme sa chceli celou touto konštrukciou vyhnúť a poškodilo by to všetky záruky bezpečnosti.

Zmena, obnova alebo zrušenie hesla, takisto ako aj zánik používateľa sa budú riešiť cez administrátora systému. Administrátor nie je zamestnancom cloudu, je to dôveryhodný zamestnanec firmy. Ten má uloženú záložnú databázu. Vždy, keď si chce používateľ zmeniť heslo, komunikuje a administrátorom. Potrdenie zmeny hesla či potvrdenie zabudnutia hesla bude prebiehať 2-faktorovou autentifikáciou, a to buď dodatočne emailom alebo sms správou, keď používateľ bude musieť do časového limitu zadať kód získaný takouto formou. Následne využijeme schému spoločného tajomstva (z ang. „secret sharing scheme“) s (n, n) treshold-om (to znamená, že n účastníkov z n všetkých účastníkov je potrebných na znovuzískanie kľúča) :

Nech s je reťazec dĺžky l zložený z núl a jednotiek, $s \in \{0,1\}^l$. Nech každý s_i je náhodný reťazec núl a jednotiek dĺžky l pre $i = 1, .., (n-1)$ a $s_n = s \oplus s_1 \oplus s_2 \oplus .... \oplus s_{n-1}$. Potom rekonštrukcia pôvodného s sa vypočíta ako $s = s_1 \oplus s_2 \oplus .... \oplus s_n$.
Táto konštrukcia spĺňa bezpečnostný predpoklad, že ľubovoľných $(n-1)$ alebo menej účastníkov pri spojení nevie zistiť absolútne nič o reťazci s. Táto schéma je perfektná schéma. [61]

Administrátor po získaní nového kľúča s (v našom prípade sa jedná o heslo od používateľa, ktoré bolo potvrdené používateľom 2-faktorovou autentifikáciou) tento kľúč rozdelí pomocou schémy spoločného tajomstva.

$$s_1 \in \{0,1\}^n, n = 256$$

$$s_2 = s \oplus s_1$$

Heslo používateľa sa dá teda opätovne získať zložením s_1 a s_2. Jedna časť, s_1, sa pošle na cloud a druhá časť sa ponechá v počítači administrátora. Reťazec s_2, ktorý je u administrátora už nikdy neopustí jeho zariadenia a nebude ho nikam posielať. Má ho v lokálnej tabuľke len pre seba. Cloud následne dáva tieto reťazce s_2 administrátorovi opäť 2-faktorovou autentifikáciou. Takto zabránime tomu, že v počítači administrátora by boli prítomné všetky heslá a tento by bolo treba zase dôkladne chrániť a vytvárať zabezpečovacie systémy a postupy.
Tento spôsoby zálohy vyriešil nasledovné problémy:

- **Používateľ si chce zmeniť heslo** (napr. má pocit, že jeho heslo mohlo byť odhalené). Používateľ vyplní svoje aktuálne a nové heslo. Administrátor obnoví heslo používateľa z cloudu a overí či ho zadal správne, a teda je tento pokus o zmenu hesla oprávnený. Nové heslo sa akceptuje opäť pomocou 2-faktorovej autentifikácie. Následne administrátor vypýta z cloudu z tabuľky prístupu záznamy s ID číslom používateľa ako „Accessor ID“, aby ich mohol prešifrovať novým heslom a uložiť do cloudu. Ako ďalšie vytvorí z hesla reťazce s_1 a s_2 a jeden z nich opäť uloží do cloudu na príslušné miesto. Zmena hesla požívateľa je obmedzená na 3 za jeden deň, aby nedochádzalo k zneužívaniu tohto procesu na vyťaženie servera požiadavkami.

- **Používateľ zabudol heslo.** Administrátor vie obnoviť používateľovo zabudnuté heslo a pomocou 2-faktorovej autentifikácie mu dať možnosť vytvoriť si nové heslo. Následne získa dáta s jeho ID číslom z cloudu, odšifruje ich obnoveným heslom a zašifruje ich novým heslom používateľa. Pomocou jeho hesla znovu vytvorí reťazce s_1 a s_2, z ktorých jeden uloží na cloud. Zabudnutia hesla sú opäť obmedzené na 3 za deň pre jedného používateľa.

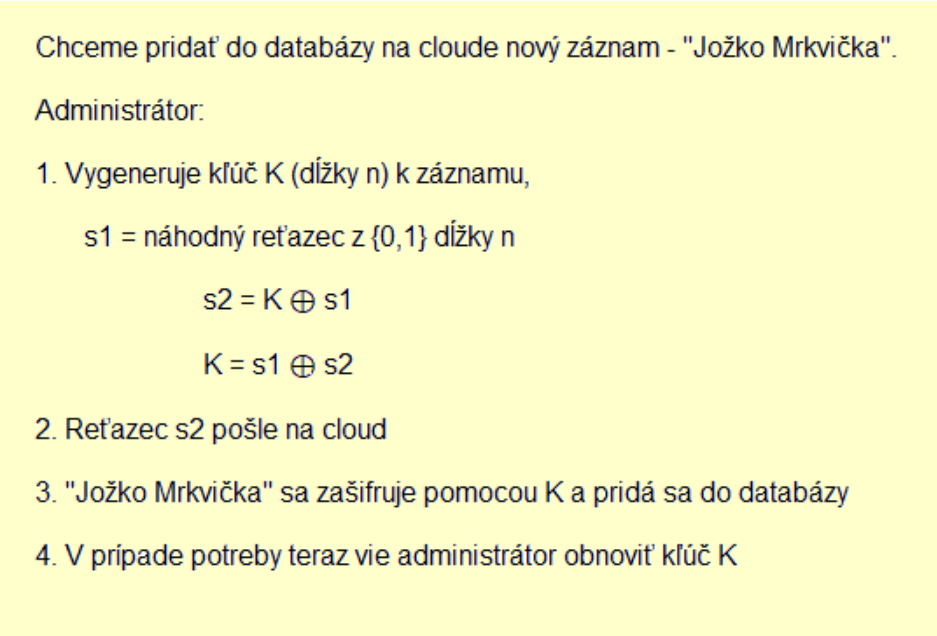
- **Používateľ nemá momentálne prístup k dátam**. Napr. bol prepustený z firmy alebo vzhľadom na jeho zdravotný stav je predpoklad, že sa v dohľadnej dobe nedostane do systému. Ak bol zároveň jediný s prístupom k týmto dátam, nikto iný ich nevie odšifrovať. Administrátor systému môže obnoviť u seba heslo používateľa, a tak prešifrovať dáta alebo kľúče k dátam v tabuľkách a prideliť ich inému používateľovi. Ak tento používateľ nebol jediným, ktorý mal prístup k daným dátam, stačí riadky s príslušným „Accessor ID“ používateľa, ktorý momentálne nemá prístup k dátam, vymazať. Bude teda prechádzať záznamy v tabuľke prístupu, ktoré majú ako „Accessor ID“ uvedené ID číslo takéhoto používateľa. Ak pod číslom „Accessed ID“ je ID číslo, ktoré sa nachádza v tabuľke

prístupu ešte aj pod iným „Accessor ID“, stačí tento záznam s „Accessor ID“ používateľa, ktorý momentálne prístup nemá, z tabuľky prístupu len vymazať. Ak takéto iné „Accessor ID“ neexistuje, mal by údaje z tabuľky prístupu zachrániť (zachrániť tak kľúč K k záznamu s príslušnými údajmi) a prideliť inému používateľovi, pretože inak by k nim nemal prístup a nemohol pristupovať nikto. To či má niekto iný prístup k danému „Accessed ID“ môže ľahko overiť vo svojej tabuľke skupín, tab. 4.2. Následne vymaže aj reťazce na obnovenie hesla daného používateľa, ktorého jeden bol uložený v cloude a jeden u administrátora.

Administrátor komunikuje s cloudom pomocou 2-faktorovej autentifikácie pri získavaní a ukladaní záložných reťazcov pre obnovu hesiel. Takisto administrátor komunikuje s používateľom pomocou 2-faktorovej autentifikácie (autentifikácia zo strany používateľa).

2-faktorová autentifikácia medzi cloudom a administrátorom sa dohodne pri uzatváraní zmluvy s cloudom, kde sa povie cloudu s kým má cloud takto komunikovať a akým spôsobom. 2-faktorová autentifikácia medzi administrátorom a používateľom sa dohaduje v čase, ked vzniká tento používateľ systému.

Z iného hľadiska môže byť heslo používateľa jeho dôverný údaj a aj keď by nemal, môže toto heslo používať na prístup aj do iných systémov. Preto by bolo vhodné, aby nemal možnosť obnoviť ho ani administrátor nášho systému. Môžeme preto touto schémou spoločného tajomstva chrániť namiesto hesiel používateľov kľúče K k jednotlivým záznamom tabuľky a ku skupinám. Obrázok 4.3. Kľúč K je teda pri vygenerovaní nového záznamu a nového K rozložený na reťazce s_1 a s_2 podľa popísanej schémy. Jeden reťazec, s_1, sa pošle na cloud, druhý uloží u administrátora. V prípade zmeny hesla sa teda len prešifruje záznam v tabuľke prístupu. V prípade zabudnutia hesla cloud zistí, ku ktorým záznamom mal prístup, obnoví k nim kľúče K a prešifruje ich pre používateľa novým heslom. V prípade, že sa stratil prístup k nejakým dátam, môže len jednoducho obnoviť ich kľúče K a zašifrovať ich heslom iného používateľa, prípadne môže vzniknúť časový úsek, v ktorom k týmto dátam nemá prístup nikto. Až v prípade potreby môže administrátor obnoviť kľúče K k týmto záznamom. Komunikácia prebieha opäť 2-faktorovou autentifikáciou.



Obr. 4.3: Možnosť obnovy kľúča K k záznamom.

4.3.4 Problém jednoduchých alebo rovnakých hesiel

Jednoduché heslá

Najväčšou slabinou pri útokoch hackerov bude práve heslo používateľov. Ľudia nie sú veľmi dobrí vo vytváraní ťažko uhádnuteľných hesiel, ale v dnešných dobách môže pomôcť riešiť tento problém password manager [10]. Zamestnanci firmy, ktorí pracujú s osobnými údajmi v cloude, by mali byť poučení ako vybrať vhodné heslo, nakoľko v našom prípade pracujú s dôležitými osobnými a citlivými údajmi, ktoré sú kryté zákonmi. My ich ale môžeme obmedziť podmienkami na heslo. V tomto prípade je najdôležitejšie dbať na dĺžku hesla. Heslo, ktoré má 6 alebo menej znakov vedia útočníci prelomiť vyskúšaním všetkých možností do pár sekúnd a je jedno či toto heslo muselo obsahovať aj čísla, malé a veľké písmená alebo dokonca aj špeciálne znaky. [25]

V roku 2012 na konferencii Password Î2 v meste Oslo predviedol Jeremi Gosney hackovanie hesiel. Spojil výpočtové sily 5 serverov, obsahujúcich 25 AMD Radeon GPU jednotiek, ktoré komunikovali medzi sebou rýchlosťou 10 Gbps. Dokázal tak hrubou silou vyskúšať všetky možnosti hesiel dĺžky 8, obsahujúcich malé a veľké písmená, čísla aj špeciálne znaky (dokopy 95 symbolov) za 5 hodín a 20 minút. Pre 7 miestne heslo by to trvalo len niečo vyše 3 minút. Toto sa mu podarilo dosiahnuť pre bezpečnostný protokol NTLM, pričom vedel vyskúšať 348 miliárd hesiel za jednu sekundu. Pre MD5 hashovanie dokázal vykonať 180 miliárd hádaní hesiel za sekundu, pre hashovanie SHA1 63 miliárd hádania hesiel. Pre hashovacie algoritmy LM, sha512crypt a bcrypt(05) to bolo 20 miliárd, 364.000 a 71.000 pokusov hesiel za sekundu. Gosney tiež povedal, že

ich prístup by bol aplikovateľný aj pre oveľa viac procesorov, napr. 128. (Odsek zo zdroja [38] upravené po update zo zdroja [51]). V roku 2020 je už možné dosiahnuť 8,5 miliardy vyskúšaných hesiel za sekundu pri hashovaní SHA1 aj jediným počítačom a behom na GPU jednotke [59]. Pre ilustráciu a porovnanie uvádzame naslednovnú tabuľku pre výpočet všetkých možných kombinácií hesiel pre rôzne dĺžky hesiel a sady:

- len písmená bez rozlišovania malých a veľkých písmen (26 symbolov)
- písmená bez rozlišovania malých a veľkých písmen + číslice (36 symbolov)
- písmená aj malé a veľké + číslice + špeciálne znaky (26+26+10+33=95 symbolov)

(v tab. 4.6). Pre príklad - pri dĺžke hesla 8 a počte rôznych symbolov 36, ktoré sme mali použiť, je počet všetkých možností takéhoto hesla 36^8.

(dĺžka, symb)	NTLM	MD5	SHA1	LM	sha512crypt	bcrypt(5)
8, 26	0,6 s	1,2 s	3,3 s	10,4 s	6,6 dňa	34 dní
8, 36	8,1 s	15,6 s	44,8 s	141 s	90 dní	460 dní
8, 95	5,3 hod	10,2 hod	29 hod	3,8 dňa	578 r	2961 r
9, 26	15,6 s	30,2 s	86,2 s	4,5 min	159 dní	885 dní
9, 36	4,9 min	9,4 min	26,9 min	1,4 hod	8,8 r	45 r
9, 95	21 dní	41 dní	116 dní	1 r	55K r	281K r
12, 26	3,2 dňa	6,1 dňa	17,5 dňa	55 dňa	8K r	42K r
12, 36	158 dní	305 dní	871 dní	7,5 r	412K r	2mil. r
12, 95	49K r	95K r	272K r	856K r	-	-

Tabuľka 4.6: Čas prelomenia hesla útokom hrubou silou, 2012, 25 GPU.
(s - sekúnd, min - minút, hod - hodín, r - rokov, K - tisíc, mil - milión)

Ľudia môžu použiť ako svoje heslo vetu zloženú z niekoľkých slov. Tu zase hrozí možnosť slovníkového útoku. Často sa používa práve slovník pre frázy Diceware, ktorý obsahuje 7776 slov (čo je 6^5, a simuluje hod 5 kockami, odtiaľ pomenovanie „diceware“) a najdlhšie slová v ňom majú 6 znakov. Pri rýchlosti hádania aj 10 biliónov ($10 * 10^{12}$) hesiel za sekundu, by sme takéto 4 slovové heslo uhádli za 6 minút, 5 slovové za 32 dní, 6 slovové za 700 rokov a 7 slovové heslo za 5 miliónov rokov. [46]

Nakoľko v našich návrhoch budeme často používať heslá používateľov ako kľúče šifier, odporúčame určiť nasledovné obmedzenia (odporúčania vychádzajú z tejto kapitoly, t.j. kap. 4.3.4 a jej zdrojov, z tab. 4.6, z kap. 4.3.5 a kap. 4.4.1):

- dĺžka hesla aspoň 8 znakov (ideálne aspoň 12 znakov),

- aspoň jedno malé písmeno,
- aspoň jedno veľké písmeno,
- aspoň jedna cifra,
- aspoň jeden špeciálny znak,
- aspoň jedno slovo (časť oddelená medzerami) nesmie byť súčasťou slovníka.

Rovnaké heslá viacerých používateľov

Ďalším problémom sú rovnaké heslá dvoch používateľov, ktorí majú prístup k rovnakým dátam. V tabuľke prístupu tab. 4.1 budú mať totožné riadky len s rozdielnym „Accessor ID“. Dá sa teda ľahko zistiť ktorí používatelia majú rovnaké heslá. Ak používateľ sám takto zistí, že niekto má rovnaké heslo ako on, môže jednoducho pristupovať ku všetkému obsahu s „Accessed ID“ obete (napr. náhodným tipovaním ID čísiel záznamov).

Preto pridáme ďalšiu malú tabuľku, kde každý používateľ bude mať vlastnú, náhodne vygenerovanú soľ, tab 4.7. Soľ taktiež zabráni predpočítaniu hashov a zašifrovaných kľúčov útočníkmi a útočník tak v jednom čase môže skúšať útok len na jedného používateľa a nie celú databázu naraz (to, ako sa táto soľ bude používať, popíšeme ďalej).

Používateľ	Soľ
(Id používateľa)	(Náhodná soľ)

Tabuľka 4.7: Tabuľka so soľou.

4.3.5 PBKDF2

Heslá používateľov môžu byť rôznych dĺžok. Ak ich však chceme použiť ako kľúče do šifrovacích funkcií, pri niektorých algoritmoch musia mať takéto kľúče pevne určenú dĺžku. Ako príklad, ďalej v texte použitá funkcia AES-256 potrebuje kľúč dlhý 256 bitov, čo je 64 znakov v hexadecimálnom kódovaní.

PBKDF2 (Password based key derivation function) je pseudonáhodná funkcia, ktorá berie ako vstup heslo používateľa, soľ, počet iterácií a požadovanú dĺžku výstupu v bajtoch a vráti reťazec požadovanej dĺžky. Počet iterácií je premenná, ktorou vieme regulovať rýchlosť útoku hrubou silou voči tejto konštrukcii. Maximálna dĺžka výstupu je $(2^{32} - 1) * H_l$, kde H_l je dĺžka výstupu použitej pseudonáhodnej funkcie. Dĺžka výstupného odvodeného kľúča je teda v podstate bez obmedzenia. [60] [36]

Funkcia: PBKDF2 (P, S, c, dkLen){}
Výstup: DK (odvodený kľúč dĺžky dkLen)

V práci už spomínaná organizáca NIST, ktorá je oficiálnou autoritou pre vydávanie štandardov, vydala odporúčania pre používanie PBKDF2. Funkcia PBKDF2 je dobrým nástrojom na sťažovanie útoku hrubou silou. Použitá pseudonáhodná funkcia vnútri PBKDF2 by mala byť HMAC, SHA-3, CMAC, KMAC, cSHAKE, ParallelHash alebo ľubovoľná hashovacia funkcia v zbierke NIST SP 800-107. Požadovaný výstup by mal byť rovnakej dĺžky ako výstup z týchto jednosmerných funkcií. Soľ by mala mať najmenej 32 bitov. Cena útoku hrubou silou závisí na počte iterácií PBKDF2, čím viac iterácií, tým dlhšie trvá vypočítať výstupný hash. Preto by počet týchto operácií mal byť tak vysoký, ako dovolí výkon servera, typicky aspoň 10.000 iterácií. [7].

Takéto spomalenie vďaka 10.000 iteráciám obmedzí možnosť hádania hesiel na 300.000 tipov za sekundu v systéme so 4 GPU jednotkami. To by znamenalo, že v tabuľke tab 4.6 sa pohybujeme okolo hodnôt v stĺpci „sha512crypt“, ktorý podobne vie vyskúšať 364.000 hesiel za sekundu. Vďaka použitiu PBKDF2 sme teda vyriešili hrozbu útoku hrubou silou a 8 miestne heslo obsahjúce malé a veľké písmena, čísla a špeciálne znaky je postačujúce. [66] [64]

4.4 Návrh 1

Tento návrh spĺňa všetky bezpodmienečné požiadavky. Podporuje aj skupiny. Kľúče však používateľ v priebehu komunikácie s cloudom z cloudu získava. Používa symetrické šifrovanie, a teda je na nastavení systému či všetky údaje v databáze sú len na čítanie a môže ich upravovať len administrátor alebo môžu dáta upravovať aj všetci používatelia, ktorí majú ku konkrétnym dátam prístup. V tejto schéme použijeme šifrovací algoritmus AES. Na overenie integrity použijeme HMAC.

4.4.1 AES

AES je symetrický (t.j. rovnaký kľúč na šifrovanie aj odšifrovanie) blokový šifrovací algoritmus schválený a certifikovaný inštitúciou NIST. Je používaný aj vládou USA na zabezpečenie tajných dát a je považovaná za jednu z najbezpečnejších symetrických šifier aké boli doposiaľ vynájdené. AES je sama o sebe len veľmi ťažko prelomiteľná ak je implementovaná korektne a je odolná voči všetkým známym útokom okrem útoku hrubou silou (bruteforce attack). Počet operácií na vykonanie útoku hrubou silou je však pri 256-bitovej AES šifre (256 bitov je dĺžka kľúča) až $3.31 * 10^{56}$, čo je číslo, ktoré je približne rovné počtu atómov vo vesmíre. AES šifrovanie je teda len tak bezpečné ako je bezpečný jeho kľúč. [10] [21]

My budeme používať AES-256 v móde CBC, čo je mód s inicializačným vektorom. Ako kľúč do AES šifry použijeme upravené užívateľské heslo, preto treba dodržať obmedzenia na heslo uvedené v kap. 4.3.4. Heslo používateľa s jeho soľou upravíme pomocou funkcie PBKDF2, spomenutej v kapitole 4.3.5. Z nej vygenerujeme 640 bitov materiálu pre kľúče. Prvých 128 bitov použijeme ako IV (inicializačný vektor) pre AES šifrovanie, ďalších 256 bitov ako kľúč do tejto šifry a posledných 256 bitov bude slúžiť neskôr na zabezpečenie/overenie integrity, teda pre autentizačný tag resp. digitálny podpis.

Šifrovanie a dešifrovanie pomocou AES je veľmi rýchle. Využitím hardvérových inštrukcií vieme s využitím jedného jadra a jedného vlákna dosiahnuť až pribilžne 1,3 cyklov/byte pre AES-128 v CBC móde pre dešifrovanie. Pre AES-256 s využitím jedného jadra a jedného vlákna je to 5,7 cyklov/byte pre šifrovanie a 1,8 cyklov/byte pre dešifrovanie. [3]

Príklad 4.1. Majme procesor s frekvenciou 3,6 GHz, čo je úplne bežné pre dnešné počítače. Tento vykoná $3.6 * 10^9$ inštrukčných cyklov za sekundu. Chceme zašifrovať 1GB dát, $1GB = 2^{30}B$. Na zašifrovanie budeme teda potrebovať $5.7 * 2^{30}$ cyklov. Zašifrovanie 1GB teda prebehne za $\frac{5,7*2^{30}}{3,6*10^9} = 1,7$ sekundy. Dešifrovanie 1GB dát prebehne za $\frac{1,8*2^{30}}{3,6*10^9} = 0,54$ sekundy.

4.4.2 HMAC

HMAC je momentálne najčastejšie používanou funkciou na autentizáciu správ. Je podobná hashovacej funkcii, vo svojom vnútri aj vybranú hashovaciu funkciu používa. Ako vstup však okrem správy berie aj tajný kľúč.

$$HMAC(k, m) = H((k \oplus opad)||H((k \oplus ipad)||m)),$$

kde k = kľúč, m = správa v plaintexte, H = hashovacia funkcia, ipad, opad = padding.

Odolnosť voči kolíziám použitej hashovacej funkcie vo vnútri funkcie HMAC nie je dôležitá a funkcia HMAC je stále bezpečná aj napriek nájdeniu kolízií. Výstup funkcie HMAC je rovnako dlhý ako výstup použitej hashovacej funkcie. [63]

Jednosmerná funkcia HMAC vytvorí hash zo správy s použitím tajného kľúča. Správny hash ku konkrétnej správe s príslušným tajným kľúčom vie vytvoriť teda len ten, čo pozná tajný kľúč, a tak vie zistiť či je zachovaná integrita správy.

4.4.3 Vlastnosti

Návrh používa tabuľku s dátami - tab. 4.5 tak, ako je uvedená. V poslednom stĺpci má digitálny podpis. Ako tabuľka prístupu je tabuľka takmer rovnaká s vyššie uvedenou

tab. 4.1. Tabuľka prístupu v návrhu 1 však ešte naviac obsahuje aj stĺpec s autentizačným tagom pre overenie či nikto nemanipuloval so záznamami v tejto tabuľke prístupu (tab. 4.8)

Accessor ID	Accessed ID	encrypted K	HMAC
(Id toho kto pristupuje)	(Id toho k čomu pristupuje)	(K zašifrované heslom toho, kto pristupuje)	(HMAC k tomuto záznamu)

Tabuľka 4.8: Tabuľka prístupu pre návrh 1.

Náhodne zvolené kľúče K pre dáta sú dĺžky 640 bitov, 128 bitov pre inicializačný vektor do AES šifrovacej funkcie, 256 bitov ako kľúč do tejto funkcie a 256 bitov ako kľúč pre autentizačný tag / do HMAC funkcie.

Náhodne zvolené kľúče/heslá skupín sú takisto dĺžky 640 bitov s rovnakým rozložením.

4.4.4 Popis komunikácie - príklad fungovania

Priebeh komunikácie používateľa s cloudom je zobrazený v obrázku 4.4.

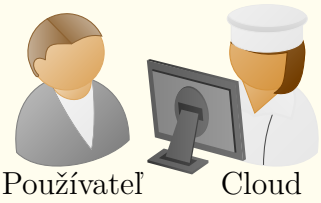
Používateľ si vypýta z cloudu svoju soľ. Taktiež si vypýta žiadané dáta. Cloud zistí, v ktorých skupinách dáta sú a v ktorých je používateľ. Podľa toho mu následne pošle vybrané záznamy z databázy - záznam z tabuľky dát ku konkrétnym dátam, záznam z tabuľky prístupu skupiny, ktorá k nim má kľúč a jeho záznam z tabuľky prístupu k tejto skupine.

Používateľ vloží svoju soľ spolu so svojim heslom do PBKDF2 funkcie, ktorá mu vráti 640 bitový reťazec. Prvých 128 bitov vloží ako inicializačný vektor do AES funkcie, ďalších 256 bitov vloží ako kľúč. Procesom dešifrovania sa dozvie ďalší 640 bitový reťazec, ktorý slúži pre prístup ku skupine. Následne overí integritu toho, čo získal - vloží tento odšifrovaný 640 bitový reťazec a posledných 256 bitov vygenerovaných z jeho hesla a soli do HMAC funkcie a výsledok porovná s posledným stĺpcom v jeho zázname.

Ako ďalšie odšifruje riadok pre skupinu rovnakým postupom. Zo získaných 640 bitov použije 128 bitov ako inicializačný vektor a ďalších 256 bitov ako kľúč do AES funkcie. Tým odšifroval kľúč K, ktorým sú zašifrované dáta a ktorý má tiež dĺžku 640 bitov. Tento si opäť môže overiť použitím posledných 256 bitov a práve získaným plaintextom v HMAC funkcii, ktorým overí integritu údajov pre skupinu.

Týmito poslednými 640 bitmi už len odšifruje dáta. Opäť použije prvých 128 bitov z tohto reťazca ako inicializačný vektor a ďalších 256 ako kľúč do AES šifry. Týmto spôsobom odšifroval dáta. Ďalej vloží dáta spolu s poslednými 256 bitmi z kľúča do

Obr. 4.4: Návrh 1, komunikácia.



Leg.:		
	id_{ja}: moje id	id_{kam}: id dát, ku ktorým chce pristúpiť
	sol: príslušná soľ	G: skupina, ktorá má prístup k dátam
	H: heslo používateľa	KG: heslo ku skupine
	K: kľúč na odšifrovanie dát	id_kam: id dát, ku ktorým chce pristúpiť

HMAC funkcie a porovná výstupný hash z posledným stĺpcom a overí tak integritu dát.

Tento postup zobrazuje zjednodušene, bez overovania integrity záznamov obr. 4.4. Ak by sme si chceli predstaviť obrázok aj s overovaním integrity, pridal by sa do obrázku len pri posielaní z cloudu aj HMAC odtlačok a v každej fáze s dešifrovaním aj následne po dešifrácii vloženie odšifrovaného textu a príslušného kľúča do HMAC funkcie a porovnanie výsledku s prijatým HMAC hashom.

4.5 Návrh 2

Pozn. autora: Tento návrh je veľmi teoretický a potreboval by ešte domyslieť niektoré vlastnosti. Je navrhnutý najmä ako náčrt, ako by sa dal riešiť problém zistenia kľúča používateľom z kap. 4.3.2. Používa šifrovanie XOR, ktoré je pri dodržaní správnych zásad teoreticky neprelomiteľné a výstup sa javí len ako skutočne náhodný reťazec šumu.

Návrh 2 rieši problém získania kľúča K zamestnancom, ktorý následne odíde z firmy alebo je vyhodený, a to tým, že k zamestnancovi sa kľúč K ani nikdy nedostane a nikdy sa ho nedozvie. Dostanú sa k nemu len priamo žiadané dáta. Nevýhodou ale je, že používa len jednoduché symetrické šifrovanie a teda práva sa nedajú rozdeliť (pre používateľov, ktorí môžu len čítať a tých, ktorí môžu aj čítať aj zapisovať). Týmto symetrickým šifrovaním je obyčajný XOR. Ďalšou nevýhodou je, že kľúče musia byť rovnakej dĺžky ako dáta.

Tento návrh ako je tu uvedený nepodporuje skupiny, a teda vychádza z tabuľky prístupu - Tab. 4.1.

Na overenie integrity používa obyčajný hash pre-xorovaný náhodným reťazcom. Na upravenie hesla používateľa používame tiež funkciu PBKDF2, uvedenú v kap. 4.3.5.

Táto metóda je vhodná ak je síce veľa záznamov v dátach, ale záznamy sú kratšie, a teda tabuľky neobsahujú príliš veľa stĺpcov, pretože sa žiadané dáta budú posielať viackrát (a preto, že kľúč musí byť rovnakej dĺžky ako dáta).

4.5.1 Vlastnosti

Upresníme vzhľad tabuľky dát z Tab. 4.5. Dáta sú zašifrované vždy náhodným reťazcom K. Každý záznam má vygenerované náhodné K dĺžky rovnakej ako je dĺžka záznamu. Uložené dáta v databáze sú následne prexorované týmto reťazcom K. Taktiež pre každý autentizačný tag pri dátach je vygenerované náhodné KH. KH má dĺžku 256 bitov a používa na zašifrovanie výstupu hashovacej funkcie. Viď tabuľku s dátami, tab. 4.9.

ID	Zašifrované dáta	Autentizačný tag
Id	$(data \oplus K)$	$(hash \oplus KH)$

Tabuľka 4.9: Návrh 2, tab. s dátami

Taktiež upravíme a upresníme tabuľku prístupu pre Návrh 2 z tab. 4.1. Uložené K v tabuľke je prexorované ošetrenou prvou časťou hesla používateľa. Uložené KH je prexorované druhou časťou oštereného hesla používateľa. Viď tab. 4.10. Heslo používateľa je upravené pomocou jeho soli a PBKDF2 funkcie na heslo dĺžky rovnakej ako dĺžka záznamu, ku ktorému pristupuje, plus dĺžka kľúča KH, čiže spolu na ((dĺžka záznamu v bitoch) + 256 bitov). Heslo H vytvorené PBKDF2 funkciou má tak 2 časti, prvá časť (označme H_1) je úsek po druhú časť, druhá časť je posledných 256 bitov (označme H_2). $H = H_1||H_2$.

Accessor ID	Accessed ID	encrypted K	encrypted KH
Kto ID	Kam ID	$(K \oplus H_1)$	$(KH \oplus H_2)$

Tabuľka 4.10: Tabuľka prístupu pre návrh 2.

4.5.2 Čítanie dát z databázy

Priebeh komunikácie medzi používateľom a cloudom je detailne zobrazený v obrázku 4.5. V návrhu sa používaju označenia:

- **nsp** - náhodný string používateľa,
- **nsc** - náhodný string cloudu,
- H - upravené heslo používateľa, ktorým sú zašifrované kľúče
- K, KH - kľúče na odšifrovanie dát a hashu
- data - odšifrované a čitateľné dáta
- hash - hash vytvorený z príslušných dát

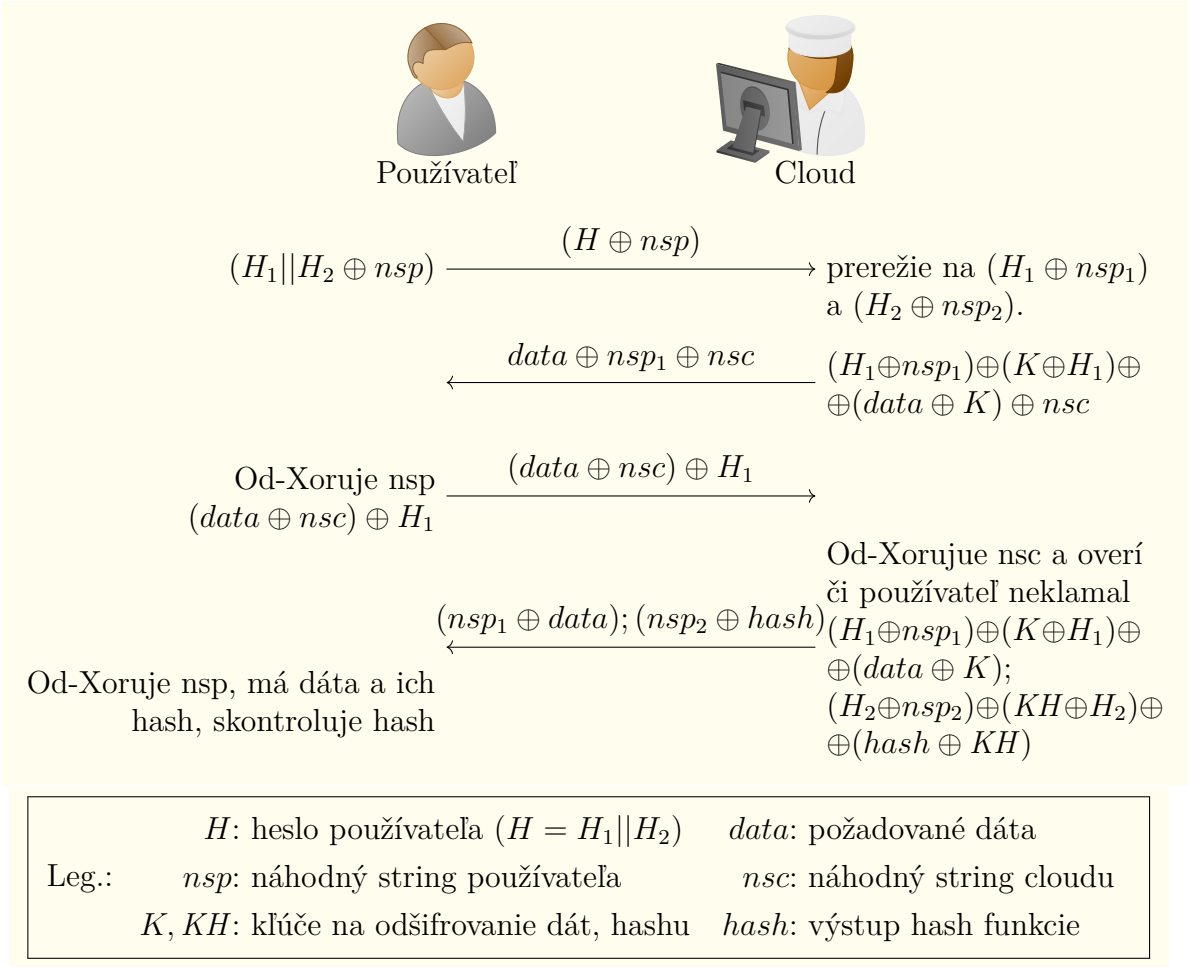
Používateľ chce získať nejaké dáta z tabuľky. Pošle teda serveru svoje Id a Id dát, ku ktorým chce pristúpiť. Server mu pošle jeho soľ a oznámi dĺžku záznamu, ku ktorému chcel pristupovať. Používateľ pomocou soli, jeho hesla, dĺžky záznamu a PBKDF2 funkcie upraví heslo na heslo dĺžky (256b + dĺžka záznamu). Toto upravené heslo H pre-xoruje náhodným reťazcom - nsp, ktorý vie len on a pošle serveru $(H \oplus nsp)$.

Server si rozdelí tento reťazec na časť pre odšifrovanie kľúča k dátam a kľúča k hashu tak, že oddelí posledných 256 bitov a ostane mu reťazec $H_2 \oplus nsp_2$. Prvá časť je $H_1 \oplus nsp_1$. Do prvej časti pri-xoruje svôj náhodný reťazec - nsc a tiež pri-xoruje doňho výraz $(K \oplus H_1)$, ktorý sa nachádza v tabuľke prístupu pod jeho Id ako Accessor ID a pod Id toho, k čomu chce pristúpiť ako Accessed ID; ďalej k nemu pri-xoruje výraz v tabuľke s dátami, ktoré majú id rovnaké ako Id toho, k čomu chce pristúpiť, a teda výraz $(data \oplus K)$. Dokopy teda zložil výraz $((H_1 \oplus nsp_1) \oplus (K \oplus H_1) \oplus (data \oplus K) \oplus nsc)$. Nsc je rovnakej dĺžky ako všetky ostatné reťazce v tomto výraze. Rovnaké výrazy sa vyxorovali na 0. To, čo vo výraze zostalo, je teda $(nsp_1 \oplus data \oplus nsc)$.

Tento reťazec dostane používateľ. Ten k nemu pri-xoruje svoje nsp_1, čím z reťazca táto hodnota vypadne a opäť k nemu pri-xoruje svoje heslo H_1. Tento výraz, teda výraz $(nsc \oplus data \oplus H_1)$ pošle na server.

Server pri-xorovaním svojho nsc z výrazu dostane výraz $(data \oplus H_1)$. S ním môže jednoducho overiť či používateľovo heslo H uvedené na začiatku komunikácie bolo správne. Ak totiž teraz pri-xoruje k výrazu, výraz z tabuľky prístupu - $(K \oplus H_1)$, mal by získať výraz $(data \oplus K)$, teda reťazec, ktorý mal uložený v tabuľke s dátami a

Obr. 4.5: Návrh 2, komunikácia.



naopak. Takto zistil či používateľ bol korektný, či neklamal a nesnažil sa zistiť údaje z databázy zadaním iných reťazcov ako svojho hesla.

Ak je všetko v poriadku, cloud pošle používateľovi naspäť údaje z prvého kroku, teda to, čo ako prvé posielal používateľ - $(H \oplus nsp)$ (prípadne mu to v tomto kroku používateľ opätovne zašle), ktoré pozná aj v rozdelenej forme na dve časti - $(H_1 \oplus nsp_1)$ a $(H_2 \oplus nsp_2)$, pri-xoruje k prvej časti údaje z tabuľky prístupu $(K \oplus H_1)$ a z tabuľky dát $(data \oplus K)$. Výraz, ktorý tak vznikne je $(H_1 \oplus nsp_1) \oplus (K \oplus H_1) \oplus (data \oplus K)$, z čoho po vypadnutí zdvojených reťazcov xorovaním ostane výraz $(data \oplus nsp_1)$. Rovnakým spôsobom pri-xoruje aj výraz na overenie hashu do druhej časti.

Používateľ teraz jednoducho, od-xorovaním nsp získa data a hash k ním. Z dát vytvorí hash a porovná so získaným hashom.

Takýto zdĺhavejší proces je potrebný. Cloud teoreticky mohol už v prvom kroku poslať používateľovi hneď požadované dáta, a teda na prvú požiadavku s $(H \oplus nsp)$ hneď reagovať zaslaním naspäť $((H_1 \oplus nsp_1) \oplus (K \oplus H_1) \oplus (data \oplus K))$, z čoho by používateľovi ostalo $(nsp_1 \oplus data)$ a mohol by hneď pristupovať k dátam. Lenže prob-

lém nastáva v prípade, že zamestnanec firmy v úlohe používateľa cloudu alebo niekto, kto pozná dáta, by mohol poslať požiadavku a tváriť sa ako hocikto iný, kto má k týmto dátam tiež prístup, ako svoje heslo by uviedol práve známe dáta a tak by mu týmto spôsobom cloud poslal nevedomky heslo tohto používateľa. S týmto heslom by ďalej mohol pristupovať ku všetkému, k čomu mal tento používateľ s prelomenou identitou prístup. Náš uvedený prístup tento problém ale nemá. Čo však môže byť nevýhodou oproti 1-krokovej výmene je možnosť DOS útoku, keďže server si musí pamätať minimálne použité nsc pri konkrétnych dialógoch.

4.6 Návrh 3

Tento návrh spĺňa všetky bezpodmienečné podmienky. Podporuje taktiež skupiny a rozlišuje práva používateľov na čítanie a čítanie aj písanie. Využíva podobne ako 1. návrh šifrovanie AES (kap. 4.4.1) a na úpravu hesiel tiež funkciu PBKDF2 (kap. 4.3.5). Na overenie integrity a autentickosti využíva digitálny podpis.

4.6.1 Digitálny podpis

Digitálny podpis slúži na zabezpečenie autentifikácie a nepopierateľnosti autorstva - po overení vieme, že správa bola vytvorená a podpísaná len tým, kto vlastní súkromný kľúč a nikto iný ho nemohol vytvoriť. Tiež slúži na zabezpečenie integrity - správa nebola menená od vytvorenia a podpísania.

Schéma digitálneho - podpisu obr. 4.6. Majme trojicu (Gen, Sig, Vrf). Nech:

- Gen je funkcia na vytvorenie dvojice verejného a súkromného kľúča (pk, sk),
- Sig je funkcia, ktorá produkuje podpis zo správy m a súkromného kľúča sk, jej výstup označme σ

$$\sigma = Sig_{sk}(m),$$

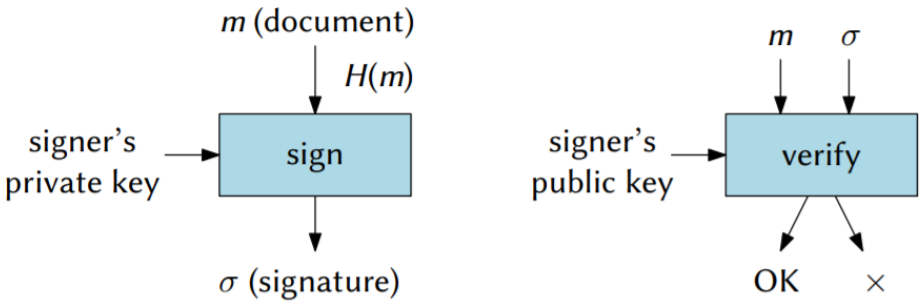
- Vrf je funkcia, ktorá zoberie správu m, podpis σ a verejný kľúč pk a overí korektnosť $Vrf_{pk}(m, \sigma)$

$$Vrf_{pk}(m, \sigma) \in \{true/OK, false/x\}.$$

Korektnosť schémy:

$$\forall (pk, sk) \leftarrow Gen(1^k) \forall m : Vrf_{pk}(m, Sig_{sk}(m)) = true$$

Táto schéma teda využíva techniky asymetrického šifrovania. Súkromným kľúčom sa digitálny podpis vytvára, verejným kľúčom sa overuje správnosť. [62]



Obr. 4.6: Schéma digitálneho podpisu. (Zdroj obrázku: [62])

Implementácií podpisových schém je viac a máme niekoľko možností, ktorú podpisovú schému použijeme v našom systéme. Odporúčame sledovať aktuálne vydané štandardy a riadiť sa nimi. [37]

4.6.2 Vlastnosti

Návrh používa tabuľku s dátami tak ako je zobrazená v tab. 4.5. Tabuľka prístupu vyzerá rovnako ako tab. 4.4. Model taktiež používa tabuľku so soľou, tab. 4.7. Pre každý záznam v tabuľke dát sú teraz generované až 3 kľúče. Jeden je 384 bitový reťazec, označme ho kľúč K, z ktorého sa 128 bitov použije ako inicializačný vektor a 256 bitov ako kľúč šifry AES-256 na zašifrovanie údajov. Ďalšie 2 kľúče sú dvojica verejného a súkromného kľúča (pk, sk) vygenerovaných funkciou Gen spomínanou v predchádzajúcej podkapitole. Pomocou funkcie Sig, súkromného kľúča a zašifrovaných údajov sa k týmto údajom vytvorí digitálny podpis. Tento podpis sa uloží vedľa záznamu. Kľúče K a pk sa zreťazia a umiestnia sa do tabuľky prístupu používateľom, resp. skupinám, ktoré majú k tomuto záznamu prístup. Súkromný kľúč na vytvorenie digitálneho podpisu sa priradí len tým, ktorí majú aj právo na zápis k tomuto záznamu. Všetky kľúče sú generované na strane prevádzkovateľa, nie cloudom.

4.6.3 Popis komunikácie

Priebeh komunikácie používateľa s cloudom je zobrazený v obrázku 4.7. Komunikácia pri čítaní dát prebieha skoro rovnako ako pri 1. návrhu.

Používateľ pošle na server požiadavku o prístup k údajom. Pošle svoje ID číslo a ID číslo záznamu, ku ktorému chce pristúpiť. Cloud zistí, do ktorých skupín patrí záznam, ku ktorému chce používateľ pristúpiť a tiež ku ktorým skupinám má používateľ prístup. Ak je takýchto záznamov viac, skúsi nájsť prvý taký, v ktorom má používateľ väčšie práva, teda práva aj na zápis (t.j. atribút K_D nie je NULL). Podľa toho mu následne

pošle príslušné záznamy v tabuľke prístupu a požadovaný záznam v tabuľke s dátami. Tiež pošle používateľovi jeho soľ z tabuľky so soľou.

Používateľ vloží získanú soľ spolu so svojim heslom do funkcie PBKDF2, ktorá mu tentokrát vráti 384 bitový reťazec. Prvých 128 bitov použije ako inicializačný vektor, ostatných 256 bitov ako kľúč do šifry AES-256. Záznam ktorý takto potom odšifruje je ten, kde je on uvedený ako „Accessor ID“ a ako „Accessed ID“ je uvedené ID grupy, cez ktorú mu cloud navrhol prístup. Ak je teda ako „Accessed ID“ uvedené ID číslo grupy, atribút K_D v tabuľke prístupu je null, pretože záznam, na ktorý odkazuje je opäť v tabuľke prístupu a ten používateľ meniť nemôže.

Odšifruje záznam. Pod kľúčom K, ktorý takto získal, je 384 bitový reťazec. Tento znovu rozdelí a prvých 128 bitov použije ako inicializačný vektor a ostatných 256 bitov ako kľúč do šifry AES-256. Záznam, ktorý takto odšifruje je ten záznam z tabuľky prístupu, ktorý mu bol poslaný, kde „Accessor ID“ je ID číslo skupiny, cez ktoré mu cloud vybral prístup a „Accessed ID“ je ID záznamu, ku ktorému chcel pôvodne pristúpiť.

Záznam odšifroval. Získal tak kľúč K zreťazený s verejným kľúčom pk na overenie podpisu. Ak má právo aj na zápis do požadovaného záznamu, získal aj súkromný kľúč na vytvorenie digitálneho podpisu. Zreťazené kľúče sú rozdelené tak, že prvých 384 bitov z kľúča predstavuje kľúč K a zvyšok je verejný kľúč pk.

Ďalším prijatým záznamom bol záznam z tabuľky s dátami, ktorý požadoval. Pomocou funkcie Vrf, verejného kľúča pk a zašifrovaných dát overí integritu a autentickosť záznamu. Ak bolo všetko v poriadku, môže dáta odšifrovať. Z kľúča K, ktorý má po rozdelení 384 bitov znovu zoberie 128 bitov ako inicializačný vektor a 256 bitov ako kľúč pre šifrovanie AES-256. Dáta, ktoré chce takto odšifrovať je konečne záznam z tabuľky dát.

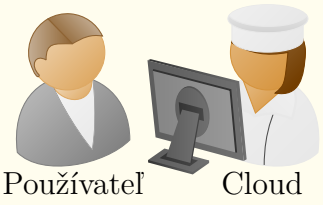
Text popisuje obrázok 4.7.

4.6.4 Zápis dát do databázy

Pri zápise dát sa zopakuje rovnaký postup ako popísaný vyššie. Nachádzame sa teda vo fáze, že používateľ má záznam, ktorý chce meniť má aj všetky potrebné kľúče - (K||pk) a sk. Používateľ upraví záznam podľa potreby, opraví, prepíše údaje v ňom. Následne použije prvých 128 bitov z kľúča K ako inicializačný vektor a ďalších 256 bitov ako kľúč do AES-256 šifry, avšak teraz spustí na tomto plaintexte proces šifrovania. Formát dát mu jeho prístupové UI meniť nedovolí, môže meniť len hodnoty vnútri atribútov. Vzniknutý zašifrovaný text a súkromný kľúč sk vloží do funkcie Sig, a tým vytvorí digitálny podpis k záznamu. Tento odošle na cloud. Cloud nahradí záznam prijatým záznamom spolu s digitálnym podpisom.

Druhou a bezpečnejšou možnosťou by bolo nechať verejné kľúče pk nezašifrované, v plaintextoch pri príslušný záznam. Mohol by si tak pre každý záznam úplne hocikto

Obr. 4.7: Návrh 3, komunikácia.



id_{ja}, id_{kam} $\xrightarrow{id_{ja}, id_{kam}}$ Cloud nájde, cez ktorú skupinu G sa k dátam vie používateľ dostať, nájde príslušnú soľ, posiela soľ a riadok z tabuľky prístupu: Kto=id_{ja}, Kam=G, kt. má prístup k id_{kam}

PBKDF2(H,soľ) odšifruje KG $\xleftarrow{sol, (KG)_{AES_H}}$

odšifruje K pomocou získaného KG $\xleftarrow{(K||pk)_{AES_{KG}}, (\varnothing/(sk)_{AES_{KG}})}$ posiela riadok z tabuľky prístupu: Kto=G, Kam=id_{kam}

overí DigP pomocou pk a $(data)_{AES_K}$, odšifruje dáta pomocou K, $\xleftarrow{(data)_{AES_K}, (\text{DigP})}$ posiela riadok z tabuľky dát: id=id_{kam}

Leg.:	id_{ja}: moje id	id_{kam}: id dát, ku ktorým chce pristúpiť
	sol: príslušná soľ	G: skupina, ktorá má prístup k dátam
	H: heslo používateľa	KG: heslo ku skupine
	K: kľúč na odšifrovanie dát	id_kam: id dát, ku ktorým chce pristúpiť
	DigP: digitálny podpis	$\varnothing$: NULL ak nemôže zapisovať

overiť či je zachovaná integrita a autentickosť záznamu. Výhodou tohto prístupu je, že by si to vedel overiť aj samotný cloud pri požiadavke o zmenu údajov/zápis. Ak používateľ pošle požiadavku na úpravu nejakého záznamu, posiela ho zašifrovaný a popísaný digitálnym podpisom. Cloud si overí digitálny podpis, verejný kľúč bude mať takto k danému záznamu k dispozícii. Digitálny podpis vytvárame v návrhu so zašifrovanými dátami. Ak overenie digitálneho podpisu bolo úspešné, cloud aktualizuje záznam na tento nový, od používateľa. Jediný ten, kto pozná súkromný kľúč digitálneho podpisu pre konkrétny záznam, a teda mal v tabuľke prístupu povolený aj zápis, vie korektný digitálny podpis vytvoriť. Ak je overenie digitálneho podpisu neúspešné, cloud zamietne zapísať zaslané dáta do databázy.

4.6.5 Vymazenie záznamu

Postup pri potrebe vymazania záznamu s osobnými údajmi niektorej dotknutej osoby je jednoduchý. Administrátor vymaže záznam s jej ID číslom. Následne pozrie do tabuľky prístupu a vymaže všetky záznamy, v ktorých bolo jej ID uvedené ako „Accessed ID“. Ak grupa, ku ktorej bol tento záznam priradený, už neobsahuje ďalšie prístupy do databázy s dátami, vymaže aj všetky záznamy z tabuľky prístupu, kde je ako „Accessed ID“ uvedená táto grupa. Obidve lokácie môže veľmi ľahko nájsť vďaka tabuľke skupín, tab 4.2.

4.7 Vylepšenie efektívnosti

V našom modeli sa snažíme v prvom rade dostatočne chrániť dáta. Každý záznam má teda vygenerovaný iný kľúč K na šifrovanie, preto vyhľadávanie v takejto databáze je príliš zdĺhavé a komplikované. Ak používateľ nevie ID číslo záznamu, ku ktorému chce pristúpiť a chce vyhľadať záznam podľa mena, musel by si postupne vypýtať zo serveru všetky záznamy, ku ktorým má prístup a postupne každý odšifrovať.

Aby bolo tento systém možné aj prakticky použiť, navrhli sme vylepšenia efektívnosti rôznych dotazov nad databázou. Zároveň sme pre toto vylepšenie funkcionality nepotrebovali znížiť bezpečnosť dát.

Pozn.: Ak sú v databáze osobné údaje dotknutých osôb a my chceme, aby si každá dotknutá osoba vedela o sebe zobraziť svoje údaje, ako je tomu tak napríklad aj v AiSe pri študentoch, tak táto osoba sa v našom pojímaní neberie ako používateľ, len ako pristupovateľ. Tento pristupovateľ nebude mať vlastné vyhľadávacie tabuľky, pretože môže pristupovať len k svojim údajom a teda svoje ID uvádza aj ako „Accessor ID“ aj ako „Accessed ID“.

4.7.1 Spoločná vyhľadávacia tabuľka

V databáze vytvoríme tabuľku, podľa toho, na akom atribúte chceme vyhľadávať.

Príklad 4.2. Chceme vyhľadávať podľa priezvisk. Vytvoríme tabuľku s 2 stĺpcami. V ľavom stĺpci budú postupne všetky možné kombinácie 3 písmenkových reťazcov - A, AA, AAA, AAB, AAC, ... ZZZ. V pravom stĺpci sú ID čísla k ľuďom, ktorých priezvisko začína na danú kombináciu písmen, tab 4.11.

Kombinácia písmen	Zoznam ID
A	{ 5, 243, 87 }
AA	{ 2, 7960 }
AAA	{ 12, 394 }
...	...
ZZZ	{ 86 }

Tabuľka 4.11: Tabuľka pre efektívnejšie vyhľadávanie (prezviská).

Takáto tabuľka ale prezrádza príliš veľa informácií. ID čísla v pravom stĺpci budú preto zašifrované. Zašifrované budú heslom používateľa, ktorý má k nim prístup. (Heslá sú bezpečné a upravené pre potreby šifry pomocou PBKDF2 tak, ako to bolo spomínané pri Návrhu 1 v kap 4.4.) Ak má takýto prístup viacero používateľov, bude viacero záznamov v tabuľke, napr. k študentovi s priezviskom Novotný a ID číslom 123 majú mať prístup 3 učitelia - U1, U2, U3. V tabuľke budú preto nasledovné záznamy:

NOV	$\{(123)_{U1}, (123)_{U2}, (123)_{U3}\}$

Útočník takto nevie zistiť, ktorý záznam má priezvisko začínajúce na NOV a ani ich počet. To, čo by vedel povedať je, ak by napr. pod kombináciou NOV neexistoval žiadny záznam a pravý stĺpec by bol prázdny, že v databáze nemáme žiadnu osobu, ktorej priezvisko by začínalo na NOV. Tomuto sa ale môžme tiež vyhnúť ak primiešame do tabuľky náhodný šum. Tento šum by boli náhodné nezmyselné reťazce rozmiestnené náhodne po celej tabuľke. Prípadne pre každý riadok by sa vygeneroval náhodný počet takýchto zašumení - 1-10 s tým, že pravdepodobnosť toho, že sa vygeneruje len práve 1 náhodný záznam je veľmi malá. Tak by sa zabránilo tomu, že by útočník vedel povedať, kto sa v databáze nenachádza.

To čo teda máme, je tabuľka, kde vľavo sú kombinácie písmen a vpravo náhodne postupnosti znakov (či už zašifrované ID čísla alebo šum). Ak chce teraz používateľ vyhľadať niekoho v databáze podľa priezviska, zadá žiadanú kombináciu písmen. Algoritmus zoberie z tabuľky všetky reťazce príslušného riadku a skúsi ich všetky postupne odšifrovať. Ak sa reťazec odšifroval na zmysluplné ID číslo, môže to byť to správne a vyskúša sa pozrieť do databázy či je to naozaj tak a až teraz robí zdĺhavejší proces odšifrovávania niekoľkých blokov textu. Mohlo sa mu ale podariť náhodou, že sa reťazec odšifroval na nejaké číslo, ktoré by mohlo byť ID. V tomto prípade po poslaní požiadavky na server o toto ID mu server buď povie, že k tomto záznam prístup nemá, pretože ho nenájde v tabuľke prístupu alebo ak zhodou okolností vyšlo ID číslo k záznamu, ku ktorému prístup má, tak mu ho zašle a používateľ u seba zistí, že to nie je záznam, ktorý hľadal. Pravdepodonosť toho, že by z reťazca odšifroval ID číslo ak sa mu to podariť nemalo je však veľmi malá.

Ak sa používateľovi v riadku s hľadanou kombináciou písmen nepodarilo odšifrovať žiadny reťazec na rozumné ID číslo, znamená to, že takýto záznam na danú kombináciu buď neexistuje alebo k nemu nemá tento používateľ prístup.

Tabuľky na uľahčenie vyhľadávania podobného charakteru môžme vytvoriť aj na ľubovoľný iný atribút (stĺpec).

Príklad 4.3. Chceme vyhľadávat žiakov podľa predmetov. V tabuľke v ľavom stĺpci teda budú názvy všetkých predmetov. V prípade, že nechceme prezradiť útočníkovi ani názvy predmetov, môžu tu byť len skratky predmetov alebo opäť začiatky názvov predmetov. V pravom stĺpci budú zašifrované ID čísla študentov, ktorý majú požadovaný predmet zapísaný. V prípade, že chceme znemožniť útočníkom aj zistiť predmety, na ktoré nikto nechodí alebo zistiť maximálne počty ľudí na predmete, môžme opäť pridať náhodný šum, tab 4.12.

Predmety	Zoznam ID
TKAP3	$\{(5)_{u1}, (243)_{u1}, (87)_{u5}, (gds)_{xx}\}$
UAV1	$\{(12)_{u5}, (sffd)_{xyd}, (394)_{u7}\}$
...	...
VaMS	$\{(86)_{u2}, (354)_{u2}, (2345)_{u2}\}$

Tabuľka 4.12: Tabuľka pre efektívnejšie vyhľadávanie (predmety).

Rovnako môžme vytvoriť tabuľku aj pre dátumy narodení. V prípade čísel to nemusia byť konkrétne hodnoty a môžeme si do ľavého stĺpca vložiť aj žiadané intervaly.

Spôsob šifrovania ID čísiel pomocou používateľských hesiel a vhodnej šifry je pri vytváraní týchto tabuliek bezpečný, pretože ak by sa podarilo útočníkovi prelomiť heslo používateľa mohol by úplne rovnako a jednoducho pristupovať k jeho záznamom a nie len využívať tieto tabuľky na útok analýzou predvypočítaných hodnôt.

Je však potrebné doplniť, že pre každú tabuľku by sa mala použiť iná spoločná soľ, pretože inak by mohol útočník zistiť korelácie medzi tabuľkami. Taktiež pri tabuľkách, kde sa očakáva, že jedna dotknutá osoba bude mať záznamy o sebe vo viacerých riadkoch v jednej vyhľadávacej tabuľke, ako je tomu tak pri tabuľke pre predmety (tab. 4.12), kde ten istý študent môže byť uvedený pri viacerých predmetoch a zašifrovaný tým istým učiteľom, ak predmet vyučuje rovnaký učiteľ, je vhodné použiť inú spoločnú soľ pre každý riadok.

Tento spôsob tiež pomôže riešiť dotazy typu "študenti starší ako xx rokov"či "študenti s lepšou známkou ako C z predmetu". Dáta v tabuľke sú zoradené podľa ľavého stĺpca a teda je jasné, ktoré majú väčšiu alebo menšiu hodnotu ako požadovaná hodnota.

4.7.2 Osobná vyhľadávacia tabuľka

V oblasti zdravotníctva alebo podobnej, pomocná spoločná tabuľka na uľahčenie vyhľadávania nemá veľmi zmysel. Zdravotníci majú v prípade potreby možnosť pristupovať k zdravotným záznamom všetkých ľudí, avšak ku svojej činnosti bežne potrebujú len užší okruh údajov o osobách.

Na Slovensku v roku 2009 bolo (z približne 18.000 všetkých lekárov) 2331 registrovaných všeobecných (obvodných) lekárov, ktorí mali mať starostlivosť o 4,2 mil. dospelých (-500.000 neregistrovaných alebo žijúcich v zahraničí). Z toho pripadá na jedného všeobecného lekára približne 1600 osôb. Optimálny počet pacientov na jedného lekára bol v tomto roku stanovený na 1500. V roku 2018 má praktický lekár 1600 až 1700 pacientov. [40] [68]

Zefektívniť vyhľadávanie na pár milisekúnd mu teda stačí len pre týchto 1600 pacientov, ktorí spadajú pod jeho služby. Každý lekár môže mať teda svoju vlastnú tabuľku týkajúcu sa len pacientov, ktorých má na starosti. Túto menšiu tabuľku s indexami osôb podľa mena alebo dátumu narodenia môže mať uloženú aj u seba lokálne alebo v nato vyhradenej časti v cloude. V oboch prípadoch sú záznamy v tabuľke zašifrované. Ak je vyhľadávacia tabuľka určená len pre jedného používateľa, môžu byť zašifrované aj hodnoty na ľavej strane.

Ak lekári hľadajú najčastejšie pacientov v databáze podľa mena, priezviska a dátumu narodenia, môžu si vytvoriť k tomu takéto tri vyhľadávacie tabuľky so svojimi pacientami. Všetky tabľky majú rozdielnu soľ. Ak by zadali všetky tieto tri parametre pre hľadanie naraz, výsledné ID sa zoberie ako prienik ID čísel z týchto troch hľadaní.

Záver

Téma ochrany osobných údajov v cloude je stále aktuálnejšia. Každým rokom narastá počet používateľov a firiem, ktorí chcú mať svoje dáta uložené v cloude. Môžu za to jeho početné a rozmanité výhody. Tieto výhody by ušetrili firme financie a starosti. Veľa firiem však ešte stále váha v prenose svojich dát na cloud. Medzi hlavné dôvody, pre ktorý sa tak ešte nerozhodli, patrí strach o bezpečnosť takýchto dát a nekompatibilita takéhoto spracovávania so zákonmi krajiny, nakoľko až 18% dát nahrávaných na cloud sú práve citlivé osobné údaje. Správa našich tajných údajov cloudom, ktorý vlastní k týmto údajom kľúč, nepripadá do úvahy. Taktiež nemáme zaručené, že cloud by takto „nešpehoval“ naše údaje a nevyužíval ich napr. na reklamné alebo štatistické účely. Naopak, lokálne šifrovanie všetkých dát u seba a len „odloženie“ na cloud je tiež neakceptovateľné, pretože je príliš neefektívne a zdĺhavé.

Pri uchovávaní a spracovávaní osobných údajov treba u nás postupovať podľa zákonov. O povinnostiach prevádzkovateľa a právach dotknutej osoby hovorí Zákon o ochrane osobných údajov, ktorý preberá regularizácie z GDPR navrhnuté Európskou úniou. Tieto zákony netreba brať na ľahkú váhu, pretože sú nastavené prísne a za ich nedodržanie hrozia vysoké pokuty.

V našej práci sme teda popísali vlastnosti cloudu. Uviedli sme rôzne výhody cloudových systémov a rozobrali sme si rôzne štatistiky a zistenia, ktoré vypovedajú o miere a spôsobe využívania cloudov. V druhej kapitole sme si rozobrali Zákon o ochrane osobných údajov, ktorý máme na Slovensku a z neho vyplývajúce práva a povinnosti pri spracovávaní osobných údajov.

V ďalšej kapitole sme si rozobrali vybrané existujúce riešenia pre ochranu dát v cloude. Boli nimi: homomorfické šifrovanie, CryptDB, C-SDA, GhostDB a spomenuli sme aj knižnice, ktoré využívajú homomorfické šifrovanie. Týmto knižniciam sme sa však ďalej nevenovali a odkázali na ďalšiu literatúru.

Homomorfikcé šifrovanie sa snaží v prvom rade riešiť bezpečnosť údajov, a to lokálnym šifrovaním. Snaží sa poskytnúť možnosť používateľom vykonávať operácie a rôzne funkcie priamo na zašifrovanom texte tak, aby výsledok bol rovnaký ako výsledok tejto operácie alebo funkcie na otvorenom texte. Toto riešenie však je v dnešných dobách neefektívne aj keď programátori usilovne pracujú na rôznych vylepšeniach tohto systému. Taktiež je problémové využitie homomorfického šifrovania v systéme s viacerými

používateľmi.

Ďalším systémom bol CryptDB. Tento pracuje na rôznych úrovniach pre rôzne atribúty dát. Pre atribúty s najväčšou mierou bezpečnosti zachováva bezpečné šifrovanie, ale neumožňuje takmer žiadne operácie nad týmito dátami. Postupne znižuje úrovne bezpečnosti a zvyšuje funkcionalitu až pre atribúty, ktorých ukrytie pred útočníkmi nie je takmer vôbec potrebné, kde používa úplne jednoduché, deterministické šifrovanie, ktoré naviac ešte aj zachovávava poradie hodnôt rovnaké s poradím zašifrovaných hodnôt. Toto riešenie je príliš špecifické a nedá sa prispôsobiť na osobné údaje. Taktiež podľa zákonov, ak vieme identifikovať osobu, tak všetky údaje, ktoré sa jej týkajú, sú jej osobné údaje, preto aj tie musia byť v tomto prípade dôsledne zabezpečené.

Ďalšie systémy - C-SDA a GhostDB využívali na svoje fungovanie špeciálny hardvér. Zatiaľ čo pri C-SDA treba hardvér priniesť na stranu servera, kde inteligentná karta vykonáva dotazy od používateľa tak, že dotaz je rozdelený na poddotazy a podľa potreby prístupu sa poddotazy vykonávajú buď u užívateľa, na karte alebo priamo v databáze. Pri GhostDB to bol hardvér na strane klienta, ktorý mal súkromnú časť databázy uloženú priamo v sebe. Obe tieto riešenia sú obmedzené výkonom a pamäťou prídavného hardvéru, ktorý musí vykonávať a spracovávať všetky dotazy. Obe tak obmedzujú používateľov v možnosti naplno využiť výhody cloudu. C-SDA dokonca na serveri šifruje dáta deterministickou symetrickou šifrou, čo nie je vhodný nápad vzhľadom na možné útoky analýzou dát. Pri CryptDB je zase problém prístupu viacerých používateľov a hrozba straty či ukradnutia fyzického zariadenia.

Poslednou kapitolou bola kapitola, v ktorej sme sa pokúšali navrhnúť vlastné riešenia.

Oproti existujúcim riešeniam sme sa v našom riešení sústredili v prvom rade na bezpečnosť a využiteľnosť riešenia pre prístup viacerých používateľov s tým, že každý požívateľ môže mať prístup k rôznej skupine dát. Taktiež oproti iným riešeniam, sme sa snažili navrhnúť systém, ktorý by sa v čo najmenšej miere spoliehal na zabezpečenie údajov nejakým programom, pretože vieme, že naprogramovať väčší systém bez rôznych chýb, bugov a zraniteľností je takmer nemožné. Naše návrhy sa spoliehajú v prvom rade na šifrovanie samotné, a to, že útočník si pozmení dotaz alebo využije chybu cloudu, aby mu napr. poslal žiadané údaje z cloudu, mu nijako nepomôže, pretože ich nebude vedieť odšifrovať. Použitím iného kľúča ku každému záznamu sa nám tiež podarilo obmedziť útoky analýzou dát.

V tejto poslednej kapitole sme teda najprv spísali požiadavky na vlastnosti systému, ktoré sú dané zo zákona. Ďalej sme spísali voliteľné požiadavky a problémy, ktoré by sme chceli riešiť a ktoré by nám mohli stáť v ceste pri navrhnutí nášho systému. Ku každej požiadavke a každému problému sme potom v krátkej úvahe rozobrali jej podstatu a uviedli náčrt jej zabezpečenia alebo riešenia. Po ujasnení týchto požiadaviek na systém sme prešli k samotným návrhom, ktoré využívali vyššie uvedené úvahy a

postupy.

Nakoniec sme v tejto kapitole navrhli bezpečné riešenie efektívnejšieho vyhľadávania v databáze podľa rôznych atribútov.

Cieľom práce bolo preskúmať a analyzovať existujúce možnosti a navrhnúť vhodné riešenie, ktoré by zabezpečilo kryptografickú ochranu osobných údajov spracovávaných v cloude. Tieto ciele sme naplnili.

Čo však ostáva otvorené pre ďalšiu prácu je analýza efektívnosti našich návrhov pri použití v reálnych systémoch. Taktisto by sa mohli skúsiť tieto návrhy implementovať a uviesť do praxe.

Veľa ľudí si neuvedomuje, že cloudové služby môžu mať veľký význam a výhody práve v oblasti bezpečnosti dát. A to hlavne z toho pohľadu, že menšia firma si nemôže dovoliť platiť bezpečnostných expertov na ochranu a analýzu interakcií s ich dátami u nich vo firme a ktorí budú reagovať na neustále vznikajúce hrozby. Cloudová firma orientujúca sa na uchovávanie práve citlivých údajov by si ale takýchto expertov platiť mohla a tým by si vytvárala výhody pred konkurenciou.

Literatúra

[1] Zákon č. 18/2018 z. z., zákon o ochrane osobných údajov a o zmene a doplnení niektorých zákonov. 2018. https://www.zakonypreludi.sk/zz/2018-18.

[2] Why did intelligence backdoors get into ibm groupware? Dec 23, 2019. https://gigazine.net/gsc_news/en/20191223-lotus-notes-nsa-backdoor/.

[3] Kahraman D. Akdemir, Martin Dixon, Wajdi K. Feghali, Patrick Fay, Vinodh Gopal, Jim Guilford, Erdinc Ozturk, Gil Wolrich, and Ronen Zohar. Breakthrough aes performance with intel ® aes new instructions. 2010.

[4] Mohamed Alloghani, Mohammed M. Alani, Dhiya Al-Jumeily, Thar Baker, Jamila Mustafina, Abir Hussain, and Ahmed J. Aljaaf. A systematic review on the status and progress of homomorphic encryption technologies. *J. Inf. Secur. Appl.*, 48, 2019. https://doi.org/10.1016/j.jisa.2019.102362.

[5] Nicolas Anciaux, Mehdi Benzine, Luc Bouganim, Philippe Pucheral, and Dennis Shasha. Ghostdb: Querying visible and hidden data without leaks. In *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data*, SIGMOD '07, page 677–688, New York, NY, USA, 2007. Association for Computing Machinery. https://doi.org/10.1145/1247480.1247555.

[6] Anastasios Arampatzis. Homomorphic encryption: What is it and how is it used. January 22, 2020. https://www.venafi.com/blog/homomorphic-encryption-what-it-and-how-it-used.

[7] Kolektív autorov. Digital identity guidelines. Technical Report 800-63B, National Institute of Standards and Technology (NIST). https://pages.nist.gov/800-63-3/sp800-63b.html.

[8] Mark Lee Badger, Timothy Grance, Robert Patt-Corner, and Jeffery M Voas. Cloud computing synopsis and recommendations. Technical report, 2012. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-146.pdf.

[9] Luc Bouganim and Philippe Pucheral. Chip-secured data access: Confidential data on untrusted servers. In *Proceedings of the 28th International Conference on Very Large Data Bases*, VLDB '02, page 131–142. VLDB Endowment, 2002. https://dl.acm.org/doi/10.5555/1287369.1287382.

[10] Contel Bradford. 5 common encryption algorithms and the unbreakables of the future. 31 July, 2014. https://blog.storagecraft.com/5-common-encryption-algorithms/.

[11] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*, ITCS '12, page 309–325, New York, NY, USA, 2012. Association for Computing Machinery. https://doi.org/10.1145/2090236.2090262.

[12] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. (leveled) fully homomorphic encryption without bootstrapping. *ACM Trans. Comput. Theory*, 6(3), July 2014. https://doi.org/10.1145/2633600.

[13] Mary Branscombe. Is homomorphic encryption ready to deliver confidential cloud computing to enterprises? July 29, 2019. https://www.techrepublic.com/article/is-homomorphic-encryption-ready-to-deliver-confidential-cloud-computing-to-enterprises/.

[14] Thomas Brewster. How a 1200-year-old hacking technique can already crack tomorrow's encrypted vaults. 3 Sep, 2015. https://www.forbes.com/sites/thomasbrewster/2015/09/03/microsoft-dumb-attacks-cracks-next-gen-cryptography/.

[15] Richard Chirgwin. Ibm's homomorphic encryption accelerated to run 75 times faster. March 8, 2018. https://www.theregister.co.uk/2018/03/08/ibm_faster_homomorphic_encryption/.

[16] Cameron Coles. 11 advantages of cloud computing and how your business can benefit from them. June 9, 2015. https://www.skyhighnetworks.com/cloud-security-blog/11-advantages-of-cloud-computing-and-how-your-business-can-benefit-from-them/.

[17] Skyhigh: Cameron Coles. Report: 18.1% of all files uploaded to the cloud contain sensitive data. December 21, 2016. https://www.skyhighnetworks.com/cloud-security-blog/report-18-1-of-all-files-uploaded-to-the-cloud-contain-sensitive-data/.

[18] Skyhigh: Cameron Coles. Only 9.4% of cloud providers are encrypting data at rest. July 16, 2015. https://www.skyhighnetworks.com/cloud-security-blog/only-9-4-of-cloud-providers-are-encrypting-data-at-rest/.

[19] Skyhigh: Cameron Coles. The average european company signs up for more than 1 new cloud service a day. July 2, 2015. https://www.skyhighnetworks.com/cloud-security-blog/the-average-european-company-signs-up-for-more-than-1-new-cloud-service-a-day/.

[20] Skyhigh: Cameron Coles. 6 cloud security issues that businesses experience. July 30, 2015. https://www.skyhighnetworks.com/cloud-security-blog/6-cloud-security-issues-that-businesses-experience/.

[21] Douglas Crawford. How does aes encryption work? 4 Feb, 2019. https://proprivacy.com/guides/aes-encryption.

[22] Paul Barton Dr Stephanie Starling Duncan Geere David McCandless, Tom Evans. World's biggest data breaches & hacks. 1 April, 2020. https://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/.

[23] Ziynet Dayıoğlu. Secure database in cloud computing - cryptdb revisited. 2014. https://pdfs.semanticscholar.org/8971/99f3350b19803f9e3a60384117bd3c1d01f8.pdf?_ga=2.136739376.1692888294.1589570818-571571802.1587123946.

[24] Ameesh Divatia. The fact and fiction of homomorphic encryption. January 1, 2019. https://www.darkreading.com/attacks-breaches/the-fact-and-fiction-of-homomorphic-encryption/a/d-id/1333691.

[25] Paul Ducklin. How to pick a proper password. 01 OCT 2014. https://nakedsecurity.sophos.com/2014/10/01/how-to-pick-a-proper-password/.

[26] Laurence Goasduff Gartner Inc.: Katie Costello. Gartner says worldwide iaas public cloud services market grew 31.3% in 2018. July 29, 2019. https://www.gartner.com/en/newsroom/press-releases/2019-07-29-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31point3-percent-in-2018.

[27] Rob van der Meulen Gartner Inc.: Katie Costello. Gartner says worldwide iaas public cloud services market grew 29.5 percent in 2017. August 1, 2018. https://www.gartner.com/en/newsroom/press-releases/2018-08-01-

gartner-says-worldwide-iaas-public-cloud-services-market-grew-30-percent-in-2017.

[28] Craig Gentry. *A Fully Homomorphic Encryption Scheme*. PhD thesis, Stanford, CA, USA, 2009. AAI3382729.

[29] Nick Gonella. Homomorphic encryption. 7 May, 2017. [Prednáška], https://www.youtube.com/watch?v=8iamgFEanZ8.

[30] Jan Hajny, Lukas Malina, Zdenek Martinasek, and Ondrej Tethal. Performance evaluation of primitives for privacy-enhancing cryptography on current smart-cards and smart-phones. In *Revised Selected Papers of the 8th International Workshop on Data Privacy Management and Autonomous Spontaneous Security - Volume 8247*, page 17–33, Berlin, Heidelberg, 2013. Springer-Verlag. https://doi.org/10.1007/978-3-642-54568-9_2.

[31] Shai Halevi and Victor Shoup. Algorithms in helib. Cryptology ePrint Archive, Report 2014/106, 2014. https://eprint.iacr.org/2014/106.

[32] İhsan Haluk AKIN and Berk Sunar. On the difficulty of securing web applications using cryptdb. Cryptology ePrint Archive, Report 2015/082, 2015. https://eprint.iacr.org/2015/082.

[33] Vincent Hu. General access control guidance for cloud systems. Technical report, 2020. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-210-draft.pdf.

[34] Gartner Inc. Gartner says worldwide iaas public cloud services market grew 31 percent in 2016. September 27, 2017. https://www.gartner.com/en/newsroom/press-releases/2017-09-27-gartner-says-worldwide-iaas-public-cloud-services-market-grew-31-percent-in-2016.

[35] Wayne Jansen and Timothy Grance. Sp 800-144. guidelines on security and privacy in public cloud computing. Technical report, 2011. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-144.pdf.

[36] B. Kaliski a A. Rusch. K. Moriarty. Pkcs #5: Password-based cryptography specification, version 2.1. Jan 2017. https://tools.ietf.org/html/rfc8018.

[37] Cameron F. Kerry, Acting Secretary, and Charles Romine Director. Fips pub 186-4 federal information processing standards publication digital signature standard (dss), 2013. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

[38] András Keszthelyi. About passwords. *Acta Polytechnica Hungarica*, 10:99–118, 01 2013.

[39] Skyhigh: Ajmal Kohgadai. 12 must-know statistics on cloud usage in the enterprise. March 9, 2017. `https://www.skyhighnetworks.com/cloud-security-blog/12-must-know-statistics-on-cloud-usage-in-the-enterprise/`.

[40] MUDr. Peter Lipták. Všeobecný praktický lekár kalkulácie. 7 Aug, 2008. `http://www.vpl.sk/sk/kalkulacie/`.

[41] Aranha Diego F. M. R. Alves, Pedro G. A framework for searching encrypted databases. *Journal of Internet Services and Applications*, 9, 3 Jan, 2018. `https://doi.org/10.1186/s13174-017-0073-0`.

[42] Malwarebytes. Backdoor. `https://www.malwarebytes.com/backdoor/`.

[43] Peter Mell and Timothy Grance. The nist definition of cloud computing. Technical Report 800-145, National Institute of Standards and Technology (NIST), Gaithersburg, MD, September 2011. `http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf`.

[44] Gartner Inc.: Susan Moore. Gartner says worldwide cloud infrastructure-as-a-service spending to grow 32.8 percent in 2015. May 18, 2015. `https://www.gartner.com/en/newsroom/press-releases/2015-05-18-gartner-says-worldwide-cloud-infrastructure-as-a-service-spending-to-grow-32-percent-in-2015`.

[45] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *Proceedings of the 17th International Conference on Theory and Application of Cryptographic Techniques*, EUROCRYPT'99, page 223–238, Berlin, Heidelberg, 1999. Springer-Verlag. `https://link.springer.com/content/pdf/10.1007%2F3-540-48910-X_16.pdf`.

[46] PasswordBits. How long should my passwords be? 8 Feb 2019. `https://passwordbits.com/password-how-long/`.

[47] Raluca Ada Popa. Cryptdb: Processing queries on an encrypted database. 28 July, 2016. [Prednáška], `https://www.youtube.com/watch?v=xsaXMUelOEA`.

[48] Raluca Ada Popa, Catherine M. S. Redfield, Nickolai Zeldovich, and Hari Balakrishnan. Cryptdb: protecting confidentiality with encrypted query processing. In *Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles*, SOSP '11, pages 85–100, New York, NY, USA, 2011. ACM. `http://doi.acm.org/10.1145/2043556.2043566`.

[49] B. Selva Rani. A novice's perception of partial homomorphic encryption schemes. *Indian journal of science and technology*, 9, 2016.

[50] ÚRAD NA OCHRANU OSOBNÝCH ÚDAJOV SLOVENSKEJ REPUBLIKY. Metodické usmernenie č. 3/2016 Cloudové služby z pohľadu zákona o ochrane osobných údajov. https://dataprotection.gov.sk/uoou/sites/default/files/mu_3_2016_cloudove_sluzby_z_pohladu_zako_na_o_ochrane_osobnych_udajov_pdf.pdf.

[51] Paul Roberts. Update: New 25 gpu monster devours passwords in seconds. 04 Dec 2012. https://securityledger.com/2012/12/new-25-gpu-monster-devours-passwords-in-seconds/.

[52] Leif Ryge. Most software already has a "golden key" backdoor: the system update. Feb 27, 2016. https://arstechnica.com/information-technology/2016/02/most-software-already-has-a-golden-key-backdoor-its-called-auto-update/.

[53] Kazue Sako, Steve Schneider, and Peter Y. A. Ryan, editors. *Computer Security - ESORICS 2019 - 24th European Symposium on Research in Computer Security, Luxembourg, September 23-27, 2019, Proceedings, Part I*, volume 11735 of *Lecture Notes in Computer Science*. Springer, 2019. [Strana 404], https://doi.org/10.1007/978-3-030-29959-0.

[54] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya. A review of homomorphic encryption libraries for secure computation. *arXiv:1812.02428*, 2018. https://www.researchgate.net/publication/329488102_A_Review_of_Homomorphic_Encryption_Libraries_for_Secure_Computation.

[55] Jaydip Sen. Homomorphic encryption: Theory & applications. *CoRR*, abs/1305.5886, 2013. http://arxiv.org/abs/1305.5886.

[56] Michael Skiba. Analysis of encrypted databases with cryptdb, 9 July, 2015. https://www.nds.ruhr-uni-bochum.de/media/ei/arbeiten/2015/10/26/thesis.pdf.

[57] Vincent Slieker. Protecting personal data in the cloud. Master's thesis, Radbound University Nijmegen, 24 August, 2015. [link, ktorý prácu hneď stiahne:] https://www.ru.nl/publish/pages/769526/z_master_thesis_vincent_slieker.pdf.

[58] Craig Smith. 70 Amazing eBay Statistics and Facts. 2018. https://expandedramblings.com/index.php/ebay-stats/.

[59] Martin Stanek. Cryptology 1, passwords. 2019/2020. [Prednáška, slide 12], http://www.dcs.fmph.uniba.sk/~stanek/l13-passwords.pdf.

[60] Martin Stanek. Cryptology 1, passwords. 2019/2020. [Prednáška, slide 21-22], http://www.dcs.fmph.uniba.sk/~stanek/l13-passwords.pdf.

[61] Martin Stanek. Cryptology(1), random stuff. 2019/2020. [Prednáška, slide 5], http://www.dcs.fmph.uniba.sk/~stanek/l30-stuff.pdf.

[62] Martin Stanek. Digital signature schemes. 2019/2020. [Prednáška], http://www.dcs.fmph.uniba.sk/~stanek/l14-digsig.pdf.

[63] Martin Stanek. Kryptológia 2, verzia 2a. September 2019. http://www.dcs.fmph.uniba.sk/~stanek/K2_v2a.pdf.

[64] Graham Steel. Parameter choice for pbkdf2. November 10, 2015. https://cryptosense.com/blog/parameter-choice-for-pbkdf2/.

[65] Lenka Stúpalová. Dôsledky zmien v kryptografických algoritmoch, 2017. http://www.dcs.fmph.uniba.sk/bakalarky/obhajene/getfile.php/mainV2.pdf?id=354&fid=690&type=application%2Fpdf.

[66] 1Password Support. How pbkdf2 strengthens your master password. April 24, 2020. https://support.1password.com/pbkdf2/.

[67] Dan Swinhoe. The 15 biggest data breaches of the 21st century. 2020. https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html.

[68] Tatiana Šušková. Čo pacienti nevidia... 25 Jun, 2018. https://zivotbezantibiotik.sk/co-pacienti-nevidia/.

[69] Rahul Varshneya. There's no such thing as a bug-free app. October 22, 2015. https://www.entrepreneur.com/article/251742.

[70] Dan Virgillito. 7 most common application backdoors. September 9, 2019. https://resources.infosecinstitute.com/7-most-common-application-backdoors/.

[71] vydavateľstvo IDC. Soc ve světě cloudů. *Security World (časopis)*, pages 26, 27, Červen, 2/2019.

[72] Wikipedia. List of data breaches, 2020. [Accessed 28-April-2020], https://en.wikipedia.org/wiki/List_of_data_breaches.

Prílohy

Na ďalšej strane príloh je Osvedčenie o absolvovaní školenia Zákon o ochrane osobných údajov. Takéto školenie musí absolvovať každá zodpovedná osoba vo firme, v ktorej sa pracuje s osobnými údajmi, resp. každá osoba, ktorá pracuje s osobnými údajmi.

ECON CONSULTING obchodno-vzdelávacia agentúra, s.r.o.
P.O. Hviezdoslava č. 396/36, 922 42 Madunice, IČO: 44978961 , DIČ: 2022953042

OSVEDČENIE

o absolvovaní školenia

Zákon o ochrane osobných údajov

Meno účastníka:

Bc. Lenka Stúpalová

Účastník sa zúčastnil školenia, ktoré svojim obsahom zodpovedá **Zákonu č. 18/2018 o ochrane osobných údajov** účinnému od 25.5.2018 a získal vedomosti potrebné pre osobu, ktorá spracováva osobné údaje.

Osvedčenie vydáva **ECON CONSULTING obchodno-vzdelávacia agentúra, s.r.o.**

V **Bratislave** dňa **28.6.2018**

Ing. Edita Svoreňová
lektor