Kryptografická ochrana osobných údajov v cloude

Lenka Stúpalová

Školiteľ: Michal Rjaško

Cloud computing

- NIST definícia

- Mnohé výhody:
  - celosvetová prístupnosť, flexibilnosť (zdroje, miesto, cena), mobilita (prezentácie), serverové miestnosti - ich zabezpečenie a správa, update raz, menší dopad na živ. prostredie – menej datacentier a efektívnejšie využívané, => zníženie nákladov firmy

- Dec 2016 - Skyhigh (McAfee) – 30 mil. zamestnancov, 600 firmách

18.1%

OF FILES IN THE CLOUD

CONTAIN SENSITIVE DATA

Ochrana osobných údajov

- 25.mája 2018 - GDPR
- OÚ – všetko, podľa čoho vie niekto nejakú osobu identifikovať a všetky údaje týkajúce sa už identifikovanej osoby

- Zabezpečenie trvalej dôvernosti, integrity, dostupnosti a odolonosti systémov spracúvania osobných údajov

- §9 Zásada správnosti.
  - OÚ musia byť správne a podľa potreby aktualizované
  - OÚ nesprávne bez odkladu vymazať alebo opraviť (prípadne označiť, že sú zlé)
- §11 Zásada integrity a dôvernosti – primeraná bezpečnosť OÚ
- §12 Zásada zodpovednosti – prevádzkovateľ zodpovedný (sankcie)
- §21 Právo na prístup k OÚ – povinnosť poskytnutia spracovávaných OÚ na vyžiadanie dotknutej osoby (opakované môže spoplatniť)
- §22 Právo na opravu OÚ, §23 Právo na výmaz OÚ

- Čiže potrebné funkcie, ktoré musíme zo zákona vedieť robiť s OÚ – nájsť/vyhľadať osobu a všetky údaje o nej, zmeniť, vymazať,..

Zábrany firiem voči cloudu

Skyhigh – viac ako 200 IT a IT security lídrov:

- 73% - bezpečnosť dát
- 38% - nekompatibilita cloudu so zákonmi krajiny a pokuty

Cieľ práce

- Preskúmať a analyzovať existujúce možnosti
- Navrhnúť vhodné riešenie zabezpečujúce kryptografickú ochranu osobných údajov spracovávaných v cloude

Hlavná požiadavka

Využiť výhody cloudu na sprac. osobných údajov.

- Dôvernosť dát
- Kompatibilita so zákonmi

- Cloud nemá prístup ku kľúču:
  - dôveryhodní zamestnanci cloudu (pr. USA), SW bugy, backdoory (NSA, Apple), krajina umiestnenia servera, data breaches (Yahoo)

- Ako teda použiť cloud na spracovanie osobných údajov?

- Lokálne šifrovanie – len ja mám kľúč
  (takto bežné aj pre tajnejšie informácie)

Nevýhoda lokálneho šifrovania

- Prístup k dátam/nájsť/zmeniť - napr. databáza ľudí, zmena priezviska/bydliska/poisťovne/oprava preklepu
  - Stiahnuť celý súbor
  - Odšifrovať celý súbor
  - Vykonať zmenu/akciu
  - Zašifrovať celý súbor
  - Poslať/nahrať na cloud

- 1000 záznamov ľudí možno cca 20MB,
  miliarda záznamov –TBs (pr. eBay)

Existujúce riešenia

- Homomorfické šifrovanie
- CryptDB
- C-SDA
- GhostDB
- …

Homomorfizmus

- Univerzum a operácia (R,*)
- Funkcia f pr. f(x)=|x|
- Vlastnosť **f(x*y) = f(x)*f(y)**

- Príklady homomorfizmov
  - (R,*): $f(x) = x^2$
  - (Z, +): f(x) = 3x
  - (Z mod p, +): f(x)= x + p

Homomorfické šifrovanie (zjednodušene)

- Štvorica (K, E, D, A) pravdepodobnostných polynomiálnych algoritmov
- Pravdepodobnosť $[D(k, c) \neq m]$ je zanedbateľná
- Homomorfická vlastnosť – A – vstup k_e, c_1, c_2, výstup c_3, nech $m_3 = m_1 \odot m_2$ (c_1, c_2 sú zašifrované m_1 a m_2), potom Pravdepodobnosť $[D(A(k_e, c_1, c_2))] \neq m_3$ je zanedbateľná

- Čiastočne a plne homomorfický kryptosystém

- Výhoda – RLWE – odolné voči kvantovým pc

Homomorfické šifrovanie

- Univerzum a operácie : (Z mod 2, +) && (Z mod 2, *)
  - T.j. množina {0,1} - bity

- $E(k,M_1 * M_2) = E(k, M_1) * E(k, M_2)$
- $E(k,M_1 + M_2) = E(k, M_1) + E(k, M_2)$

- Skúsme zašifrovať (1 + a*b)

  Enc(k, 1+a*b)= Enc(k, 1) + Enc(k, a)* Enc(k, b)

a	b	1 + a*b
0	0	1
0	1	1
1	0	1
1	1	0

- Máme NAND
- NAND bloky –> ľubovoľné logické hradlo - logický obvod -> ľubovoľná funkcia -> ľubovoľný program

  (Každý program vieme zredukovať až na logické hradlá)

- Spravíme program z NANDov, zašifrujeme, pošleme na cloud, cloud vykoná neznámu funkciu na neznámych dátach a vráti výstup

  (napr. nájdenie a zmena údaju)

Prečo teda nemáme HŠ všade?

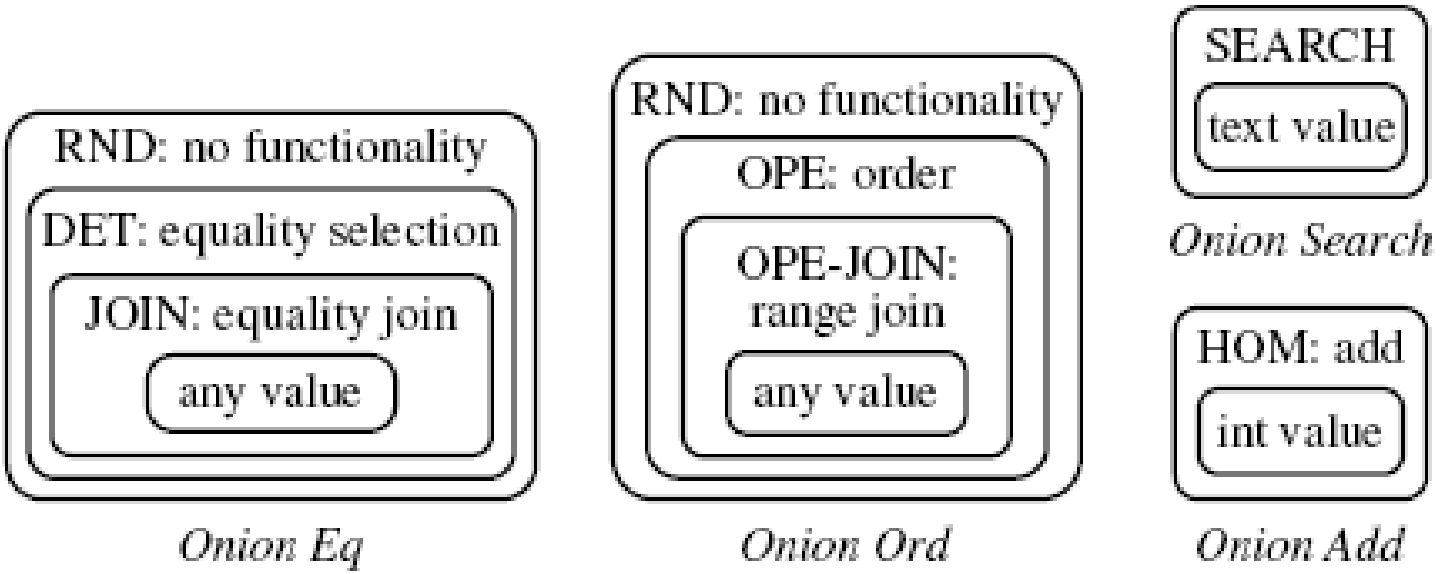
- Nepraktické
- Veľkosť programu len z NAND operácií
- Prvý nárvh schémy – 2009 Gentry +IBM - biliónkrát pomalší ako operácie na plaintexte
- 2013 – zlepšenie na milión – 16 jadrový server – ak trvala operácia na plaintexte 1s, s hom. šif. bude trvať 12 dní
- 2018 – zrýchlenie 15-75 krát

- Nepodporuje prístup viacerých používateľov

CryptDB

- Pracuje v rôznych vrstvách - bezpečnosť vs funkcionalita
- Rôzne stĺpce majú rôznu mieru zabezpečenia a funkcionality
- Ako treba vykonávať nad databázou dotazy, tak prešifruje do nižších vrstiev

- Najvyššia vrstva – RND – bez funkcionality


RND: no functionality
DET: equality selection
JOIN: equality join
any value
Onion Eq
RND: no functionality
OPE: order
OPE-JOIN:
range join
any value
Onion Ord
SEARCH
text value
Onion Search
HOM: add
int value
Onion Add

CryptDB

- Ďalšie vrstvy:
  - HOM – čiast. homomorfické šifr. - SUM, AVG
  - SEARCH – len celé slová, veľmi obmedzené
  - DET – porovnávanie s reťazcom – equality selection
  - OPE – porovnávanie poradia medzi sebou
  - JOIN, OPE-JOIN – vie robiť JOIN tabuliek

Nevýhody CryptDB

- Špecifické
- Veľkosť tabuliek – 2 stĺpce môžu mať aj 8 stĺpcov v CryptDB
- CryptDB je akoby proxy medzi používateľom a cloudom – tretia str.

- Potvrdené rôzne útoky na prístup k dátam, útok analýzou dát

C-SDA

- Chip secured data access
- Inteligentná karta na strane cloudu
- Rozdeľuje dotaz na poddotazy – server, karta, klient
  - Server – ako DET v CryptDB
  - Karta – hom. op., agregácia, šifrovanie a odšifrovanie, väčšie práva ako klient
  - Klinet – len časti dotazu s prezentáciou výsledku (sort, filtry,..)

Nevýhody C-SDA

- Karta – obmedzená, komplexné dotazy neefektívne, malá RAM, nevyužívanie cloudu plnohodnotne
- Potreba umiestnenia karty na strane cloudu
- Ľahké útoky analýzou dát kvôli deterministickému šifrovaniu

GhostDB

- Inteligentný HW na strane klienta, USB
- Databáza – verejná a súkromná, súkromná časť na HW

- Vie vykonať všetky SQL dotazy
- Ak v dotaze netreba verejné dáta, ku cloudu ani nepristupuje

Nevýhody GhostDB

- Odcudzenie, strata HW – strata dát
- Nevyužitie všetkých výhod cloudu – RAM, CPU na HW
- Prístup viacerých používateľov naraz

Návrh riešenia

Použité algorimty

- AES-256
  - Používaný vládou US, považovaný za najbezpečnejšiu šifru
  - Odolná voči všetkým známym útokom okrem útoku hrubou silou
  - Kľúč 256 bitov

- PBKDF2
  - Pseudonáhodná funkcia
  - Umožňuje použiť heslo používateľa ako kľúče do šifier
  - Vyrobí z hesla a soli potrebný počet bitov

- Použitie hesla a soli používateľa ako kľúč do AES pomocou PBKDF2

Princíp riešenia

Princíp riešenia

- Každý záznam v tabuľke zašifrovaný náhodným K

Id záznamu / študenta	Meno, priezvisko,	Rodné číslo...
nezašifrované	ostatné tu všetko zašifrované náhodným K pre každý záznam	

Tabuľka prístupu

Accessor ID Kto pristupuje	Accessed ID Kam pristupuje	K k danému riadku zašifrované jeho heslom

- Každý záznam v tabuľke zašifrovaný náhodným K
- K v tabuľke prístupu zašifrované heslom používateľa
- Deterministické šifrovanie, ale každý záznam má iné K

Integrita a autentickosť

- Do tabuľky s dátami – stĺpec s digitálnym podpisom

Id študenta	Meno, priezvisko	Rodné číslo...	Digitálny podpis / hash
nezašifrované	ostatné tu všetko zašifrované náhodným K pre každý záznam		HMAC, dig. podpis

Integrita a autentickosť

- Do tabuľky s dátami – stĺpec s digitálnym podpisom
- Do tabuľky prístupu – všetci môžu písať – súčasť kľúča K
  – niekto môže aj písať – stĺpec pre kľúč na DP

Accessor ID Kto pristupuje	Accessed ID Kam pristupuje	K||pk k danému riadku zašifrované jeho heslom	sk pre vytvorenie digitálneho podpisu k danému záznamu

Príklad

Tabuľka s dátami

Id	Meno	Priezvisko,..	Dig. podpis
12	(Lenka,	Stúpalová,...)$_K$	(záznam+sk$_1$)

Tabuľka prístupu

Accessor ID Kto pristupuje	Accessed ID Kam pristupuje	K||pk k danému riadku zašifrované jeho heslom	sk pre vytvorenie digitálneho podpisu k danému záznamu
31 (študent)	12	(K||pk)$_{passŠtudenta}$	null
48 (učiteľ)	12	(K||pk)$_{passUčiteľa}$	(sk$_1$)$_{passUčiteľa}$

Riešenie efektívnosti

- Kombinácie písmen začiatkov priezvisk, zašifrovaný zoznam ID

Kombinácia písmen	Zoznam ID
A	$\{ (5)_{p1}, (243)_{p2} , (87)_{p2}\}$
AA	…
AAA	…
…	…
ZZZ	…

Riešenie efektívnosti

- ID sú zašifrované používateľovým heslom
- Podobne tabuľky napr. pre dátumy narodení, školské predmety,....

- Pridanie šumu

Záver

- Zanalyzovali sme riešenia – upúšťali od bezpečnosti kvôli funkcionalite alebo nevyužívali výhody cloudu
- Navrhli sme riešenie, zamerané v prvom rade na bezpečnosť *všetkých* dát a súlad so zákonmi
- Možná budúca práca – implementácia a analýza efektívnosti

Ďakujem za pozornosť!

Posudok oponenta

- 1. Poznáte ISO normy ISO/IEC 27017, 27018, 27036-4, 27701, 27799 a dokumenty NIST SP 800-144 a SP 800- 210? Ak áno, prečo ste ich v práci nepoužili?

- 2. Čím je podložené tvrdenie, že 3072 bitový kľúč RSA je rovnako silný ako 128 bitový kľúč symetrickej šifry? (str. 4)

- 3. Je bloková šifra iteratívna šifra, ako tvrdíte na str. 4?

Posudok oponenta

- 4. Čo je identita, identifikátor? (str. 17)

- 5. Je dešifrovací algoritmus deterministický, keď sa pripúšťa nejednoznačnosť dešifrovania? (str. 25)

Posudok oponenta

- 6. (str. 26) vysvetlite f(x) = Evaluate(key, funkcia)
  a) Aký kľúč sa používa?

  b) Čo to je za funkciu, ktorá dokáže algoritmus (funkciu) deterministicky transformovať na logickú schému (dokonca kombinačný obvod)?

  c) Aká je zložitosť tejto funkcie a zložitosť výsledného logického obvodu?

  d) Tvrdenie, že výstupom funkcie je “séria logických NAND hradiel” je nezmysel, aby hradlá realizovali funkciu, musia tvoriť schému

Ďakujem za pozornosť!