UNIVERSITATEA POLITEHNICA BUCUREȘTI
ȘCOALA DOCTORALĂ DE AUTOMATICĂ ȘI CALCULATOARE

Rezumat
Teză de doctorat
În Calculatoare și Tehnologia Informației

Securitatea și protecția datelor private în tehnologiile actuale

Iulia-Maria Florea

Coordonator științific:
Prof. dr. ing. Răzvan-Victor Rughiniș

BUCUREŞTI

2022

Abstract

Soluțiile de păstrare a confidențialității au apărut ca o condiție prealabilă absolută pentru schimbul de informații sensibile atunci când se dezvoltă algoritmi de analiză și validare a datelor. Confidențialitatea se concentrează pe respectarea individualității, în timp ce informațiile de interes pot fi extrase din tiparele populației. Principalul lor dezavantaj al acestor metode este complexitatea lor, lipsa documentației și lipsa algoritmilor de exemplificare open-source. Pe de altă parte, mecanismele tradiționale de securitate nu reușesc să gestioneze datele mari, din cauza vitezei, varietății și volumului mare.

Prezenta teză construiește un ecosistem pentru o discuție mai complexă cu privire la fezabilitatea soluțiilor de confidențialitate în big data. Atunci când se combină soluții orientate spre cercetare cu cantități variate și mari de informații, pot fi ridicate subiecte multiple de cercetare. Unul dintre ele se referă la compromisul între disponibilitatea datelor și securitate. Un altul se referă la cei mai cunoscuți algoritmi criptografici și la aplicabilitatea lor.

Scopul nostru este să construim un sistem de procesare pas cu pas pentru big data, să înțelegem nevoia de confidențialitate la fiecare nivel și să dezvoltăm modalități prin care tehnologiile existente se potrivesc cerințelor și reglementărilor legale fără a afecta experienta utilizatorului. Am definit patru acțiuni majore în tranziția datelor: colectarea, transmiterea, stocarea și procesarea efectivă. Pentru fiecare dintre ele, am ales mai multe exemple bazate pe realitate și am dezvoltat aplicații pentru păstrarea confidențialității. Obiectivele noastre au vizat atingerea limitelor de utilizare ale soluțiilor orientate spre cercetare, dezvoltarea acestora pentru scenarii mai complexe și realizarea unor modalități mai bune de encodare a datelor pentru a maximiza atât calitatea experienței, cât și securitatea utilizatorilor.

Am început prin a explora în detaliu stiva simplificată de rețea a dispozitivelor embedded, protocoalele standard, îmbunătățirile de securitate corespunzătoare și limitările acestora. Am studiat în continuare metodele existente de criptare, autentificare și autorizare în dispozitivele cu resurse reduse. Ulterior, am propus o extindere practică a celor mai comune scenarii bazate pe IoT cu protocoale securizate.

În ceea ce privește transmiterea datelor, am decis să începem cercetarea de la unul dintre conceptele fundamentale de securitate, confidențialitate diferențială (differential privacy). Am cercetat metodele de ultimă generație de a oferi obfuscarea locației (geo-indistinguishability) și am dezvoltat o nouă metodă de a ascunde locația exactă a utilizatorilor de telefonie mobilă în orașe. Am investigat în continuare atacurile cunoscute și am testat implementarea noastră împotriva lor.

In domeniul stocarii datelor, ne-am concentrat activitatea pe una dintre cele mai interesante soluții orientate spre confidențialitate: searchable encryption. Am luat în considerare

mai multe variante (inclusiv hidden vector encryption sau inner product encryption) și am găsit scenariile cele mai potrivite pentru fiecare. Apoi, am propus metode de encodare customizate pentru a reduce costul de procesare suplimentar implicat de aplicarea tehnicilor de searchable encryption, care sunt costisitoare din punctul de vedere al resurselor utilizate.

În cele din urmă, ne-am mutat atentia către procesarea datelor și modul în care confidențialitatea poate fi implementată în algoritmii de învățare automata (machine learning). Deoarece a existat un interes tot mai mare pentru automatizare și obținerea unei intelegeri mai bune asupra datelor, am considerat acest subiect ca unul relevant în prezent. Chiar daca prezinta un interes crescut, algoritmii de învățare automată consumă multe resurse și un nivel suplimentar de securitate poate duce la erori sau rezultate nesatisfăcătoare. Am dezvoltat un algoritm de clasificare folosind o combinație între algoritmi de învățare automată și etichetare manuală folosind punctul de vedere al unui expert pentru a demonstra că informațiile pot fi obținute din orice tip de date. Astfel, am semnalat importanța confidențialității la acest nivel și am efectuat câteva experimente pentru a înțelege starea actuală a soluțiilor de păstrare a confidențialității. Am identificat principalele zone de congestie și cum efectul lor poate fi diminuat.

Am dezvoltat soluția complexă a unui proces de păstrare a confidențialității, dar, într-un domeniu vast precum big data, acest tip de soluție nu este unic, nici universal și poate fi văzut mai degrabă ca un punct de plecare pentru subiecte de cercetare mai specializate.

1 INTRODUCERE

În Decembrie 2014, un profesor de la Universitatea Cambridge a avertizat departamentul juridic al Universității cu privire la o aplicație construită de un profesor de psihologie care a colectat date a milioane de utilizatori Facebook fără știrea acestora. [1] Date de acest tip ar putea fi folosite pentru a ajunge la o profunda intelegere asupra personalităților, comportamentului și nevoilor oamenilor. Beneficiarul principal al aplicației a fost o firmă de consultanță politică puțin cunoscută, numită Cambridge Analytica [1]. Ceea ce s-a întâmplat în continuare a fost unul dintre cele mai importante scandaluri din ultimii ani: implicațiile utilizării datelor Facebook în alegerile americane din 2016, care au dus la numeroase discuții despre confidențialitatea datelor..

O urmare a acelui acelui scandal este Regulamentul General privind Protecția Datelor (GDPR), aplicat din mai 2018 în Uniunea Europeană, constând într-un set de reguli privind colectarea, stocarea și prelucrarea datelor cu caracter personal. Acestea se referă nu numai la identificatori direcți, cum ar fi numele complet sau numărul național de identitate, ci și la informații indirecte, cum ar fi numere de telefon, adrese IP sau fotografii. [2] Scopul lor este de a forța ca analiza datelor să se concentreze pe determinari statistice, fără a putea re-identifica indivizi unici, atunci când se compară setul de date anonimizate cu alte seturi similare. Reidentificarea este o problemă intalnită în confidențialitatea datelor, iar tehnicile de bază de anonimizare nu pot preveni dezanonimizari pornind de la cunoștințe anterioare. De exemplu, au existat câteva reidentificări celebre în trecut, iar una dintre ele se bazează pe setul de date Netflix. In 2006 compania a lansat un concurs pentru a găsi metode mai precise de a face recomandari de filme și, ca date de antrenare, a publicat un set de date anonimizate cu recenzii de filme de la aproape 500.000 de clienți. Identificatorii personali ai clienților au fost eliminati, iar seturile de date constau din ID-uri unice ale abonaților, evaluări ale filmelor și datele la care au fost făcute acele evaluări. O echipă de cercetători au corelat informațiile anonimizate furnizate cu baza de date publică IMDB și au putut identifica utilizatorii comparând evaluările filmelor și datele comentariilor. GDPR consideră un set de date anonimizat atunci când reidentificarea este puțin probabilă sau greu de obținut. Aspectele tehnice ale anonimizării sunt unul dintre punctele acoperite de această cercetare.

Pe lângă tehnicile slabe de anonimizare, problemele de securitate au devenit o problemă din ce în ce mai mare. Numărul atacurilor cibernetice a crescut substanțial în ultimii ani, iar costul total al criminalității cibernetice a înregistrat o rată de creștere de 15% în fiecare an. A crescut de la 3 trilioane de dolari în 2015 spre un potențial 10,5 trilioane de dolari până în 2025. De asemenea, frecvența atacurilor cibernetice a crescut de aproape 4 ori, de la una la fiecare 40 de secunde în 2016 la una la fiecare 11 secunde în 2021. [3] Confidențialitatea datelor, în acest caz, este și mai expusă riscului, deoarece atacatorii pot fura informațiile în text clar și pot infera date din mai multe surse.

De la izbucnirea pandemiei de COVID-19, criminalitatea cibernetică a crescut cu 600% [4] și 57% dintre factorii de decizie IT din companii consideră că lucrătorii la distanță pot reprezenta o amenințare suplimentară pentru securitatea internă. Misiunea lor este să construiască o soluție bazată pe confidențialitate și să încerce să adauge niște niveluri suplimentare de securitate, dacă este posibil. Prima opțiune este ca datele să fie migrate în cloud în loc să fie stocate local, mai ales pentru a avea mai multe instanțe versionate și copii de rezervă. O altă soluție ar fi asigurarea protecției tuturor datelor. Acest lucru necesită soluții holistice, pentru a se asigura că lucrătorii de la distanță au acces de oriunde la tot ceea ce au nevoie și toate informațiile trebuie să fie conforme cu reglementările locale.

Toate aceste cerințe, ilustrate în Figura 1.1, conduc la mai multe subiecte de cercetare în domeniul confidențialității, un capitol care a fost văzut mai degrabă ca academic și orientat spre cercetare, decât practic în scenariile din viața reală. Acest lucru s-a schimbat în 2016, când conceptele orientate spre confidențialitate, orientate pe conținut, precum confidențialitatea diferențială (differential privacy) [5] au fost popularizate de companii precum Google [6] sau Apple [7].

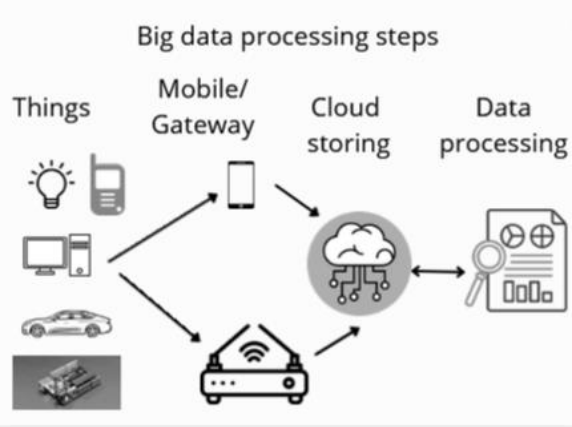

Fig. 1.1 Etapele de procesare a datelor in big data

2 SCOPUL TEZEI

2.1 Întrebări de cercetare

Cercetarea efectuată în această teză urmărește etapele principale de prelucrare a datelor: culegerea, colectarea sau transmiterea datelor, stocarea și prelucrarea datelor.

Am început prin a analiza modalitatile de colectare a datelor și, mai precis, am considerat dispozitivele IoT ca o sursă principală de noi informații. Cercetările pe această temă au fost făcute de ceva vreme, așa că eram mai degrabă interesați să înțelegem dacă securitatea oferită de protocoalele actuale era suficient de stabilă și greu de compromis în unele scenarii securizate din viața reală. Întrebarea ridicată în acest capitol este următoarea:

> **Q1**: Care sunt principalele protocoale de securitate din lumea IoT și cum pot fi combinate în scenarii ce cuprind obiecte inteligente (smart-objects)?

Al doilea pas s-a axat pe colectarea și transmiterea datelor de pe un dispozitiv mobil către un server de stocare. Deoarece dispozitivele IoT sunt mai degrabă staționare, am luat în considerare un scenariu diferit, al unui telefon mobil și un subiect arzător al zilelor noastre: confidențialitatea în timpul utilizării serviciilor bazate pe locație. Aceasta a oferit o nouă întrebare de cercetare:

> **Q2**: Este posibil să oferiți confidențialitate în timp ce utilizați servicii bazate pe locație? Care sunt limitele, principalele atacuri și cum se poate face acest lucru fără a afecta nevoile și experiența utilizatorului?

Al treilea pas se referă la stocarea datelor. Aici, ne-am concentrat pe idei mai degrabă academice, pairing based cryptography și fully homomorphic encryptions. Acestea au fost în centrul atenției în ultimii ani, dar există încă unele subiecte care necesită cercetări suplimentare.

> **Q3**: Ce tip de date pot fi stocate folosind fully homomorphic encryption? Ce tip de algoritm folosim pentru diferite tipuri de informații? Care sunt dimensiunile maxime ale bazei de date astfel încat pierderile de performanta să fie acceptabile?

Al patrulea pas se referă la prelucrarea datelor. Aici, am combinat două subiecte evidență ale lumii de astăzi, inteligența artificială și confidențialitatea. Am ales să construim un algoritm de clasificare specific pentru texte tehnice scurte și am oferit câteva răspunsuri legate de confidențialitate în algoritmii de învățare automată. Ne-am bazat cercetarea pe două întrebări diferite:

> **Q4**: Este necesară confidențialitatea în anumite scenarii specifice? Utilizatorii pot obține informații din orice tip de date, cum ar fi texte tehnice specializate, fără adnotări?

Q5: Care este penalizarea de performanță introdusă de adăugarea confidențialității în învățarea automata? Care este cel mai fezabil caz de utilizare pentru aceasta?

2.2 Contribuții

Teza își are rădăcinile în problemele de securitate și confidențialitate ale lumii din ultimii câțiva ani, începând cu numărul tot mai mare de atacuri de securitate ce se bazeaza pe dispozitive IoT, scandalurile majore care gravitează în jurul datelor oamenilor și preocuparea tot mai mare pentru protejarea confidențialității informațiilor sensibile.

Primul pas în cercetare a fost înțelegerea celor mai importante protocoale IoT, la fiecare nivel al stivei specifice de rețea. Am descoperit o stivă de rețea simplificată care conține nivelurile următoare: fizic, MAC, Rețea, Adaptare, Transport și Aplicație. Am găsit sub-nivelurile de securitate existente la fiecare nivel și ne-am concentrat pe studierea securității și interconectarii protocoalelor nivelului Aplicație. În timpul acestei etape, am găsit răspunsul la prima întrebare de cercetare:

A1: Lucrările [9] și [10] demonstrează că protocoalele de securitate relevante la nivelul Aplicație sunt DTLS și criptarea cu cheie simetrică, care poate face parte dintr-un scenariu smart-campus, orientat spre monitorizarea consumului de resurse.

Am ajuns la concluzia că acest subiect a fost explorat pe larg în ultima vreme și că metodele existente, dacă sunt implementate, sunt suficiente pentru a oferi nivelul necesar de securitate.

Pe de altă parte, studiind etapa de colectare și transmitere a datelor, am găsit un subiect legat de confidențialitate care nu a fost explorat suficient și era mai degrabă încă în zona de cercetare. Am ales să studiem confidențialitatea utilizatorilor de telefonie mobilă care trimit date către site-uri web compatibile cu HTML5 ce recepționează informații despre locație. Am pornit de la presupunerea că locația exactă nu a fost întotdeauna necesară și că riscurile de confidențialitate pot fi uneori mai mari decât beneficiile, deoarece datele despre locație pot duce la informații despre hobby-uri personale, program sau probleme de sănătate. Am urmărit a doua întrebare din secțiunea anterioară și am constatat că:

A2: Este fezabil să adăugați zgomot peste o locație exactă, chiar și să construiți o cale falsă, dar realist, într-un oraș, fără o penalizare de performanță. Lucrările din [11] și [12] arată tipul de algoritm care poate fi utilizat, scenariile în care poate fi implementat, cantitatea necesară de spațiu de stocare și puterea de procesare și principalele tipuri de atacuri care trebuie evitate.

Următoarele contribuții s-au concentrat pe etapa de stocare și au combinat un subiect de cercetare și academic, searchable encryption, cu big data. Am pornit de la diferiți algoritmi

actuali, i-am extins și adaptat pentru un anumit caz de utilizare. De asemenea, am efectuat câteva experimente legate de modul în care algoritmul existent ar funcționa pe diferite tipuri de date. Căutarea prin date criptate fără a fi nevoie să le decriptezi ar rezolva o problemă importantă de confidențialitate. Am urmărit a treia întrebare de cercetare și am oferit următorul răspuns:

A3: Fully homomorphic encryption poate fi testată mai întâi pe date numerice. Este posibil ca anumiti algoritmi care doar identifică date cu anumite proprietăți să fie extinsi, astfel incat sa le și decripteze. Penalizarea de performanță poate fi redusă prin codificarea datelor în structuri mai mici și, în funcție de proprietarul datelor și de utilizator, pot fi utilizate scheme mai rapide cu chei simetrice. Lucrarea din [13] detaliază modul în care encodarea și diferitele tipuri de căutări pot fi făcute, în timp ce lucrarea din [14] se concentrează pe găsirea celui mai potrivit algoritm pentru diferite tipuri de date numerice.

După ce datele sunt preluate și stocate, acestea pot fi utilizate pentru procesare. Majoritatea algoritmilor de învățare automată necesită adnotarea datelor de antrenare, dar ne-a interesat cum pot fi procesate datele fără adnotări. Scopul principal al confidențialității datelor este acela de a evita obținerea de informații personale despre persoane fără știrea acestora și am fost interesați să aflăm dacă acest tip de scenariu ar fi posibil. Am început prin a construi un prototip, un algoritm de clasificare pentru text tehnic scurt și am urmărit întrebarea legată de confidențialitate ridicată și mai sus. Am ajuns la următoarele răspunsuri:

A4: Lucrarea din [15] demonstrează că, în unele cazuri, algoritmii nesupravegheați de clasificare nu reușesc să ofere rezultate bune. Rulați peste texte tehnice scurte nu ofera rezultate satisfăcătoare. Cu toate acestea, acest algoritmi pot fi integrați cu succes cu unele categorii manuale definite de un expert și oferă rezultate interesante. Acest lucru demonstrează că majoritatea tipurilor de date pot oferi informații valoroase și confidențialitatea ar trebui să fie o preocupare.

A5: Lucrarea de la [16] a arătat că algoritmii de învățare automată de actuali care păstrează confidențialitatea pot fi folosiți pentru a clasifica texte. Am descoperit că penalizarea de performanță principală se întâmplă atunci când se transmit și se primesc date între un server central și lucrători privați, că acești algoritmi sunt aproape la fel de precisi ca soluțiile centralizate și că este fezabil să ruleze pe un număr mare de executori, cum ar fi smartphone-uri în timpul încărcării.

3 CULEGEREA DATELOR

Internetul lucrurilor (IoT) a devenit un domeniu important de cercetare. Poate fi descris ca un sistem de comunicare în care orice dispozitiv poate fi conectat la Internet și poate să se identifice unic față de alte obiecte. Poate fi aplicat în diferite domenii, de la uz personal, cum ar fi casele inteligente sau dispozitivele portabile până la domenii precum monitorizarea mediului urban, îngrijirea sănătății, automatizarea industrială sau situațiile de urgență.

În general, dispozitivele IoT au memorie puțină, capacitate redusă a bateriei, capacități de procesare reduse și condiții radio vulnerabile. Stiva standard TCP/IP nu este potrivită pentru acest mediu, așa că grupurile de lucru au început să actualizeze protocoalele existente la versiuni noi pentru IoT.

Ar trebui luată în considerare o schemă de adresare, cum ar fi IPv6, deoarece există miliarde de noduri interconectate. Mai multe grupuri de lucru au început deja să standardizeze protocoalele specifice IoT, cum ar fi 6LoWPAN [21] (RFC 4944 și RFC 6282) [25], IEEE802.15.4 [23] și ZigBee descriu modalități de utilizare a IPv6 în medii constrânse. Alte cerințe se referă la securitate și confidențialitate, deoarece numărul atacurilor de tip Denial of Service a crescut recent.

La nivelul Aplicație, o modalitate obișnuită de a primi și solicita date este utilizarea arhitecturii Web și, mai precis, HTTP. Aceasta utilizează URI-uri ca identificatori de resurse și se bazează pe arhitectura REST pentru a publica informații. Pentru dispozitivele embedded, există un grup de lucru IETF intitulat “Constrained RESTful Environments” (CoRE) care își propune să dezvolte protocoale RESTful compatibile cu HTTP pentru dispozitivele cu resurse limitate. Ei au descris specificația CoAP[34], un protocol de nivel de Aplicație pentru IoT.

3.1 Scenariul Smart Campus propus

O idee propusă a unui scenariu Smart Campus s-a axat pe monitorizarea confortului studenților și profesorilor în sălile de clasă și birourile clădirilor universitare. Am propus implementarea unei rețele IoT compusă din dispozitive gateway și un număr mare de noduri senzor.

Scopul acestei rețele IoT a fost de a monitoriza parametrii ambientali precum temperatura, umiditatea, presiunea, luminozitatea, calitatea aerului și prezența. Datele colectate au fost analizate folosind diverși algoritmi de detectare a evenimentelor care ar aduce disconfort studenților și profesorilor, de exemplu: luminozitate scăzută sau temperatură scăzută în timpul orelor, concentrație mare de CO_2 etc. În cazul unor astfel de evenimente, sistemul poate ajusta parametrii ambientali, de exemplu reglarea temperaturii prin sistemul de aer condiționat.

Ca orice sistem, un campus inteligent ar trebui să fie securizat. Datele nu pot circula în text clar între dispozitive, ci ar trebui să fie confidențiale și private. Primul pas pe care ne-am concentrat a fost să adăugăm securitate la nivelul dispozitivului, să criptăm comunicarea dintre noduri și serverele de stocare a datelor. La acest nivel, dispozitivele ar trebui să își dovedească identitatea, să împiedice accesul neautorizat, să semneze și să cripteze datele. În rețelele IoT, gateway-ul de asemenea translateaza date între diferite protocoale wireless, deoarece 802.11 nu este o soluție uzuală pentru dispozitivele IoT. Acest aspect ar trebui, de asemenea, luat în considerare, deoarece gateway-ul este responsabil să mențină datele integre și private atunci când se realizează translația între protocoale.

Pentru autentificare, certificatele X.509 nu pot fi folosite, deoarece dispozitivele IoT nu au suficientă memorie și putere de procesare pentru a le valida. În schimb, pot fi folosite identificarea prin frecvență radio, shared secret sau adresa MAC. DTLS este unul dintre cele mai comune protocoale de securitate, bazat pe TLS, și care oferă confidențialitate și criptare. Securitatea în CoAP , la nivelul Aplicație, se bazează adesea pe DTLS [39]. Aceasta este versiunea bazată pe UDP a TLS, concepută pentru a oferi securitate end-to-end. Oferă negociere flexibilă, folosind seturi de cifruri și mecanisme criptografice. Impactul asupra dispozitivelor cu resurse limitate este legat de handshake-ul inițial și de procesarea tuturor pachetelor securizate.

3.2 Concluziile capitolului

Protocoalele de securitate necesare pentru dispozitivele IoT sunt deja dezvoltate și munca ar trebui să se concentreze pe activarea acestora pe dispozitive cu resurse limitate. Pe baza DTLS, pot fi efectuate operațiuni customizate. Am propus o soluție cu nevoie reduse de procesare, folosind chei pre-partajate pentru arhitecturi client-server. Hardware-ul embedded acceptă derivarea cheilor cu AES-128 și, în DTLS [39], au fost adăugate două opțiuni: „AUTH” și „AUTH_MSG_TYPE”. Fiecare client are o cheie configurată, iar serverul are o listă cu toate ID-urile de client posibile. Primul mesaj trimis de la client către server conține ID-ul clientului și un token unic. Serverul a găsește parola asociată dispozitivului, derivează cheia, adăugă o variabilă aleatorie nonce_1 și trimite un mesaj criptat pe post de challenge. Când clientul primește challenge-ul, a folosește cheia partajată pentru a decripta pachetul, primește cheia și valoarea lui nonce_1 și a trimite un răspuns folosind cheia derivată și token-ul. Dacă token-ul a fost același ca în transmisia inițială, atunci clientul a fost autentificat. Payload-ul util de la nivelul Aplicație este criptat, iar opțiunile din antetul CoAP sunt folosite pentru a separa cererile criptate de cele în text clar.

A învăța despre dezvoltarea unui sistem inteligent pentru clădiri, campusuri și locuințe duce la dobândirea de cunoștințe atât despre avantajele, cât și despre problemele legate de conectarea dispozitivelor multiple și eterogene. Pe de o parte, acestea pot îmbunătăți calitatea

vieții și utilizarea resurselor, dar vin și cu riscuri de securitate și confidențialitate. Adăugarea securității la IoT este o sarcină dificilă, deoarece cele mai sigure mecanisme din stiva TCP/IP nu sunt potrivite. De asemenea, se poate folosi protocolul DTLS standard, dar in combinație cu optimizări care duc la consum mai mic de resurse, mai ales in faza de handshake inițial.

4 COLECTAREA ȘI TRANSMITEREA DATELOR

Tehnicile de ascundere a locației adresează subiectul confidențialității locației, oferind soluții bazate pe mai mulți termeni științifici împrumutați de la subiecte similare legate de confidențialitate, cum ar fi [50] și [51].

Termenul de k-anonimity[52] este adesea folosit în contextul confidențialității locației. Inițial, era legat de zona de management al datelor și statistică. Având un set de date care conțin o listă de persoane și informații de identificare despre fiecare dintre ele, k-anonimitaty este o proprietate a unui subset al acestui set de date, care afirmă că orice combinație de date din acest subset nu va mai identifica mai puțin de k persoane din subsetul menținând totuși utilitatea datelor. Printr-un algoritm de generalizare, acesta urmărește păstrarea datelor relevante în setul de date, ascunzând în același timp informații specifice. Ulterior, conceptul a fost adoptat și în domeniul confidențialității locației. De exemplu, putem presupune că avem o hartă cu puncte identic anonime care reprezintă locațiile utilizatorilor și un set de date care conține numele și coordonatele GPS ale fiecărui utilizator. Prin potrivirea informațiilor din setul de date cu harta, am fi capabili să identificăm fiecare utilizator. Algoritmul de K-anonimity transformă locațiile GPS în valori mai generale, cum ar fi strada, cartierul, orașul, țara în care se află utilizatorii. În acest fel, atunci când se încearcă punerea in corespondența a celor două surse de date, un atacator va putea deduce că un numar k de utilizatori se află în același oraș, dar nu va putea potrivi individual fiecare utilizator cu locația sa de pe hartă.

Un alt concept este confidențialitatea diferențială (differential privacy) [80]. Acesta a fost introdus în domeniul statisticii și face uz de randomizare și zgomot pentru a păstra datele anonime. Când o interogare este aplicată pe două baze de date adiacente (care au doar un rând diferit), trebuie să returneze rezultate similare. [10] oferă o definiție formală a confidențialității diferențiate. O funcție randomizată K oferă confidențialitate diferențială dacă, pentru oricare două seturi de date D1 și D2 care diferă pe un singur rând, este indeplinita condiția:

$$Pr[(K(D_1) \epsilon S] \leq exp(\epsilon) \times Pr[(K(D_2) \epsilon S] \qquad (4.1)$$

Definiția de mai sus susține că, atunci când se adaugă un nou set de date la o bază de date, informațiile ar trebui să afecteze rezultatul unei funcții randomizate aplicate peste acea bază de date la un anumit nivel, mai mic decât exp(ϵ). O valoare mai mare a lui ϵ implică mai puțină confidențialitate și mai puțin zgomot adăugat datelor, în timp ce o valoare mai mică a lui ϵ implică mai multă confidențialitate și o mai mare pierdere de calitate. Tehnicile de confidențialitate diferențială preiau datele de intrare de la utilizator și le modifică prin aplicarea unei funcții de distribuție, cum ar fi distribuția Laplace. Rezultatul acestei funcții va fi similar cu valoarea de intrare într-o anumită măsură, pentru a fi în continuare utilă pentru produsul statistic. Cu toate acestea, un atacator nu poate reconstrui datele de intrare pornind de la rezultat, păstrând astfel anonimatul utilizatorului.

În domeniul confidențialitații locației, adăugarea de zgomot la o locație precisă oferă un rezultat care poate plasa cu precizie un utilizator într-o zonă geografică dorită, ascunzând în același timp locația exactă în acea zonă. Acest lucru este util în multe scenarii, un exemplu comun fiind un serviciu meteorologic. Serviciul are nevoie de o locație generală (oraș, parte a unui oraș) pentru a furniza date meteorologice exacte, dar nu necesită o precizie a locației la nivel de stradă. Utilizatorul primește prognoza meteo în timp ce locația sa exactă nu va fi transferată pe serverul serviciului meteo, care ar putea fi susceptibil la un atac care poate dezvălui locațiile utilizatorilor.

4.1 Design-ul soluției

Vă prezentăm o soluție care obscurizează locația reală și adună informații la nivel local, astfel încât utilizatorii să poată afla despre impactul pe care îl poate avea urmărirea. Sistemul acționează ca o interfață între locația reală a utilizatorului și aplicațiile care o folosesc și este prezentat în Figura 4.1.

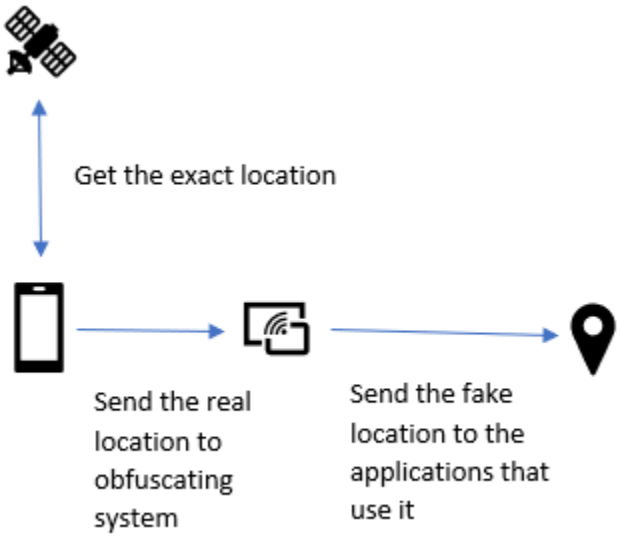

Figure 4.1: Interfață de obscurizare a locației în contextul aplicațiilor mobile

Principala cerință a sistemului este să respecte regulile de confidențialitate diferențială, ceea ce înseamnă că distanța hamming dintre locația reală și cea fabricată nu ar trebui să creeze o pierdere de calitate mai mare de ϵ. Soluția preia locația exactă, aplică algoritmul de obfuscare și apoi furnizează coordonatele fabricate.

Am extins ideea de adăugare de zgomot, deoarece algoritmii de bază s-au dovedit a fi susceptibili la diferite tipuri de atacuri. Ideea noastră principală a fost să folosim setul de date cu locații populare și am calculat posibilele locații false pe baza acestuia. Harta a fost împărțită în celule, fiecare conținând un scor de popularitate, variind de la 0 la 9. Datele de intrare necesare

au fost reprezentate de o pereche latitudine-longitudine care indică locația care urmează să fie obscurizată și nivelul de obscurizare, variind între real (nu au fost aplicate modificări), scăzut (locația fabricată a fost plasată nu departe de locația inițială), medie și înaltă (locația fabricată va fi plasată departe de locația inițială).

Apoi, locația primită ca intrare a fost plasată în perimetrul unei celule prin determinarea distanței celei mai apropiate dintre coordonatele de intrare și coordonatele celulei. După ce a găsit celula din care aparține locația inițială, a iterat peste celulele din jur și le-a marcat pe cele cu un scor de popularitate rezonabil. Apoi a selectat o celulă aleatorie din această colecție și a furnizat ca rezultat o pereche de coordonate latitudine-longitudine.

Pentru a spori și mai mult confidențialitatea, locația de ieșire nu a fost pur și simplu selectată aleatoriu, ci funcția aleatorie este o funcție ponderată. Fiecare celulă are o pondere specifică, proporțională cu scorul său. O celulă cu un scor mai mare are șanse mai mari, dar nu absolute, de a fi selectată ca rezultat.

Pentru urmărirea locației, am adăugat o secțiune care realizează logica de a determina unde ar putea fi plasată o nouă locație fabricată, pe baza locațiilor istorice ale utilizatorului. Atunci când o locație reală este procesată și este aproape de o locație falsă, aceste două locații ajută la identificarea orientării utilizatorului. Distanțele dintre locațiile false consecutive sunt proporționale cu distanțele dintre locațiile lor reale corespunzătoare, oferind mai multă plauzibilitate traseului fals.

4.2 Rezultate obținute

Pentru a testa algoritmul de urmărire, am creat un scenariu în care o persoană călătorește prin Beijing, lăsând o urmă de cinci locații reale diferite. Aceste locații au servit ca intrare atât pentru precedenta generație, cât și pentru cea actuală a acestui algoritm. Algoritmul a tratat matricea ca pe o serie de locații și fiecare rulare a apelat rezultatul ultimei rulări, făcând-o să ruleze într-o manieră stateful. Figura 4.2 prezintă toate cele trei seturi de locații. Cu marcaje galbene, am reprezentat traiectoria reală în acest scenariu. Numerele de pe fiecare marcaj arată ordinea în care au fost generați marcajele. Marcajele albaștra arată rezultatul versiunii staționare a algoritmului. Conform acestui rezultat, un atacator ar afla că utilizatorul ar fi călătorit probabil spre vest din poziția 1 și apoi spre sud-est pe o distanță considerabilă. În funcție de intervalul de timp disponibil, acesta poate duce la pierderea credibilității locației utilizatorului în fața atacatorului. Roșu marchează rezultatul algoritmului descris în această lucrare. La prima vedere, se pare că a cincea locație lipsește din setul de rezultate. De fapt, există cinci locații, dar a treia și a patra locație sunt contopite. Este imediat vizibil că traiectoria reală și cea „roșie” sunt foarte asemănătoare în ceea ce privește orientarea sau direcția, ceea ce este scopul acestui algoritm. Fiecare dintre locațiile „roșii” este plasată pe o celulă relativ populară, prin urmare locația este

plauzibilă. Un atacator fără cunoștințe prealabile despre victimă nu este capabil să distingă această urmă fabricată de o urmă reală.

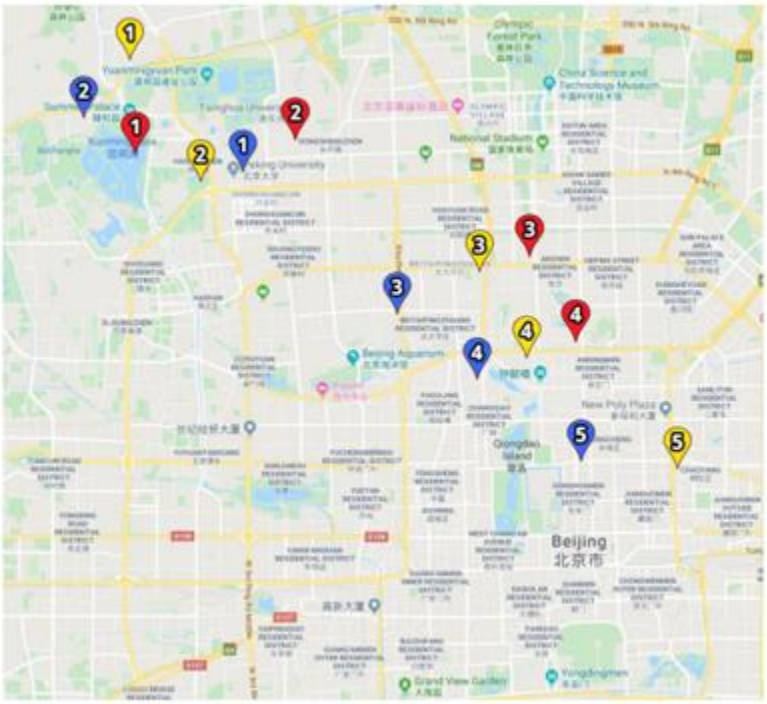

Fig.4.2 Comparația algoritmului curent cu versiunea staționară și traiectoria reală

4.2.1 Concluzii din perspectiva securitatii

Primul atac împotriva căruia a fost testat algoritmul a fost atacul de tip same-origin. A fost selectată o locație de intrare aleatorie. Această locație este staționară și a devenit sursa cererilor de obscurizare. În faza următoare, o locație fabricată a fost generată de 1000 de ori, folosind aceeași intrare. Deși lucrarea [88] afirmă că sunt necesare mai puține rulări pentru a demasca o locație reală, rezultatul devine mai relevant atunci când sunt efectuate mai multe iterații. Figura 4.3 prezintă rezultatele.

Clusterele sunt dispersate în tot orașul, fără nicio relație între poziția clusterului față de origine și densitatea clusterului. Există clustere de dimensiuni mai mari în cadranul din stânga jos, deoarece acolo sunt unele dintre cele mai populare zone. Ca regulă generală, densitatea unui cluster este proporțională doar cu popularitatea zonei respective a hărții. Ca măsură de precauție, dacă locația reală a unui utilizator se întâmplă să fie într-un loc atât de popular, nu va fi generată nicio locație falsă în acel loc. Locațiile de ieșire vor fi generate numai în locuri populare diferite de origine. Algoritmul nu este vulnerabil la atacul de tip same-origin, deoarece distribuția locațiilor generate pe hartă nu conduce către locația inițială.

Al doilea tip de atac se poate descrie ca o testare a algoritmului folosind mai multe date de intrare diferite, cu speranța de a identifica intrarea care generează o ieșire specifică. În general, atacurile de acest tip se numesc atacuri cu forță brută. Acest tip de atac este eficient în principal în cazul algoritmilor determiniști, dar într-o oarecare măsură poate fi folosit pentru a ghici zonele de intrare aproximative atunci când este desfășurat împotriva algoritmilor de obscurizare a locației. Atacatorul începe cu o anumită locație de ieșire (obscurizată). De exemplu, putem presupune că această locație de ieșire este plasată la capătul de nord al unui oraș mare. Atacatorul alimentează algoritmul cu diferite locații de intrare, încercând să genereze diverse ieșiri. Deși s-ar putea să nu poată alege între mai multe intrări aparent valide, cel mai probabil va observa că intrările aproape de capătul de sud al orașului nu generează ieșiri în nord (pentru că este prea departe). Pe baza acestei observații, el este capabil să plaseze locația reală a utilizatorului undeva spre Nord, sau spre centrul orașului, sau ușor spre Nord-Vest sau Nord-Est. Pentru o ieșire fixă dată, vom rula algoritmul cu diverse intrări și vom analiza rezultatele.

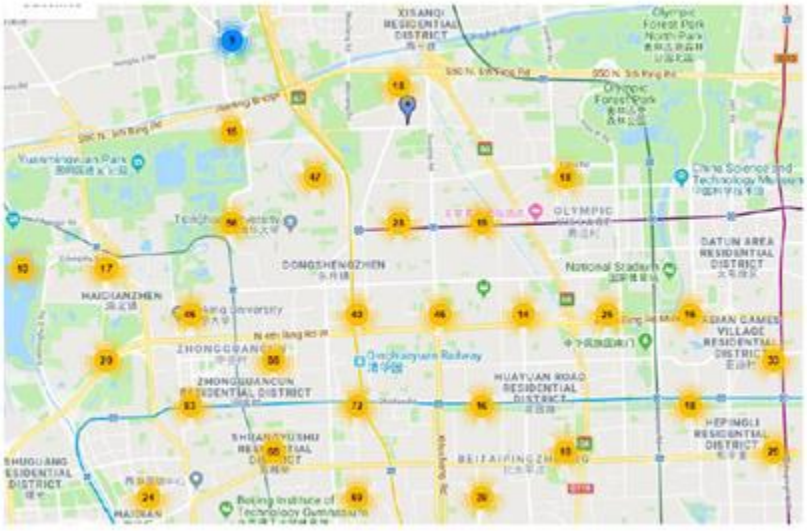

Fig. 4.3 Atac de tip “same-origin” asupra solutiei noastre de obscurizare a locației

Comportamentul algoritmului variază considerabil în funcție de poziția ieșirii desemnate. Acesta scanează o zonă limitată din jurul locației de intrare și alege o celulă aleatoare a hărții care este mai populară decât un anumit prag de popularitate. Desigur, dacă o celulă de pe hartă nu are un scor de popularitate suficient de mare, nu va fi niciodată aleasă ca rezultat. Dacă ieșirea desemnată se încadrează în oricare dintre aceste celule, acest atac eșuează deoarece nu va exista nicio intrare care ar putea genera rezultatul așteptat. Pentru cazul limită în care locația de ieșire este plasată în singurul loc foarte popular dintr-o regiune, algoritmul este de așteptat să returneze celula de pe hartă foarte des.

Într-un scenariu obișnuit, harta este probabil să conțină mai multe locații populare în zona destul de largă din jurul ieșirii desemnate. A fost testat un scenariu comun și rezultatul a fost reprezentat în Fig. 4.4. Acest grafic a fost generat cu datele colectate pe baza aceleiași hărți și acelorași locații de intrare care au fost folosite pentru tot restul experimentului. Scorul unei celule de pe hartă de care aparțin datele de intrare a fost modificat cu fiecare iterație, în timp ce toate celelalte scoruri au rămas neschimbate.

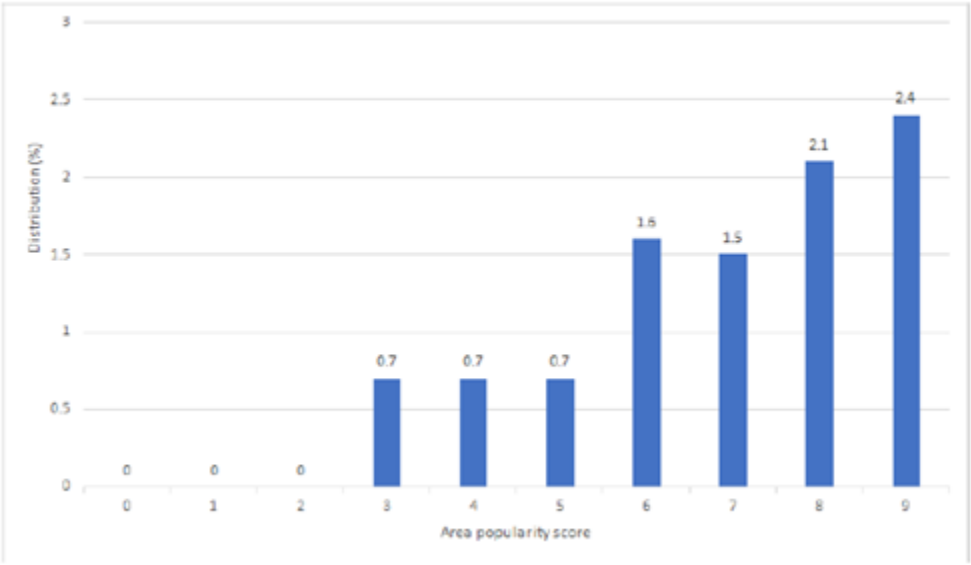

Fig.4.4 Distribuția locației in funcție de un rezultat fix

Scorurile de popularitate sunt normalizate pentru a se încadra în intervalul de la 0 la 9, 0 fiind o zonă fără interes din partea pietonilor și 9 fiind o zonă foarte populară pentru mulți cetățeni. Când scorul de popularitate al zonei din care face parte locația de intrare este mai mic decât un anumit prag, setat în prezent la 3, nu există locații generate acolo. Apoi, pe măsură ce scorul crește, unele dintre locațiile generate sunt plasate lângă locația de intrare. Deoarece harta de scoruri este destul de echilibrată, chiar și cu un punctaj maxim de 9 puncte, doar 2,4% din locații sunt plasate în zona observată. Acest echilibru este impus de regula potrivit căreia, chiar dacă o zonă are punctajul maxim, nu se garantează că va fi selectată. În schimb, are doar o șansă puțin mai mare (pondere, într-o distribuție aleatorie ponderată) în comparație cu alte zone cu scoruri de 8 sau mai mici.

5 CĂUTARE PRIN DATE CRIPTATE

În acest capitol, prezentăm două scenarii privind confidențialitatea din big data. Suntem interesați să căutăm prin date sensibile criptate, fără a fi nevoie să le decriptăm. Primul scenariu este legat de traficul de rețea, în special de traficul generat de companii, când un administrator de rețea ar fi interesat să obțină informații fără a pune în pericol confidențialitatea angajaților, iar al doilea se concentrează pe experimente cu date financiare de bază private.

5.1 Date despre traficul din rețea

Abordările pe care le prezentăm în capitol urmează două scheme diferite, prezentate în Figurile 5.1 și 5.2. În ambele scheme, există o autoritate de încredere care generează cheile. Informațiile sunt stocate pe servere externe, care pot fi „oneste, dar curioase”, iar datele sensibile trebuie protejate. Acest lucru creează nevoia de stocare a datelor criptate, în loc de text simplu, pentru a proteja confidențialitatea.

Prima soluție, din Figura 5.1, se bazează pe criptarea cheii simetrice, în care o parte, de încredere, generează cheia care va fi folosită pentru a cripta fluxurile de trafic și apoi pentru căutare. Cheia nu poate fi trimisă utilizatorilor. În acest caz, proprietarul datelor criptează datele și apoi le poate stoca pe un server extern. Când utilizatorii au nevoie de informații, acesta trimite cereri către serverul de stocare, care rulează algoritmii de căutare în date criptate și returnează rezultatele.

În al doilea caz, în Figura 5.2, autoritatea generează o pereche de chei publice și private. Utilizatorii pot obține cheia publică folosită pentru criptare și pot trimite rezultatul criptat direct la server. O entitate centrală de încredere, care deține cheia privată, va rula interogările și va decripta datele potrivite.

Având în vedere analiza fluxurilor de rețea dintr-o companie, primul caz înseamnă că toate informațiile în text clar vor fi trecute printr-un server care va cripta totul și va transmite datele către o stocare externă. În al doilea caz, jurnalele vor fi trimise criptate din diferite puncte, cum ar fi computerele utilizatorilor, dar numai un administrator va putea căuta prin ele.

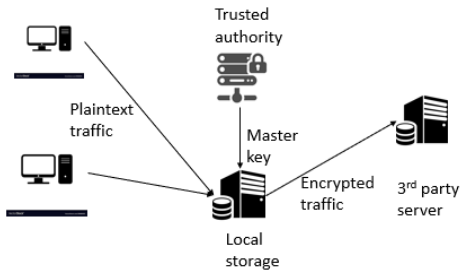

Fig. 5.1. Criptare simetrică; traficul e criptat de o autoritate de securitate

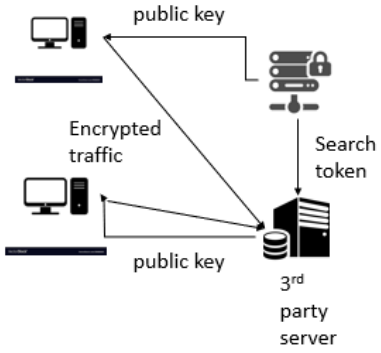

Fig. 5.2 Model de criptare asimetrică

În acest capitol, am pornit de la implementări de deja existente, pe care le-am extins și personalizat pentru traficul de rețea. Administratorul de rețea, în calitate de proprietar al datelor, a primit o cheie pentru a căuta prin date, care se potrivește cu un anumit predicat de căutare predefinit. Fluxurile de date conțineau adrese IP și porturi. Am implementat interogări de la nivel de subrețele și am extins algoritmii pentru a și decripta traficul care se potrivește condițiilor de căutare.

Un exemplu ar fi că tot traficul de rețea ar trebui să fie efectuat prin SSL/TLS și astfel să folosească portul de destinație 443. Orice trafic suspect ar putea fi considerat având portul de destinație 80. Administratorul de rețea poate seta un token de căutare pentru portul destinație 80 și atunci orice fel de flux de date suspect va fi returnat.

5.1.1 Experimente

Pentru implementările bazate pe cheii simetrice, alegem să comparăm implementarea noastră, bazată în sistemele Inner Product [91] cu o soluție numită HXT [67] și bazată pe index inversat. Pentru a prezenta diferențele de timp, alegem trei interogări care returnează 1%, 3% și 5% din baza de date.

Algoritmii au fost testați pe un server Ubuntu 16.0 care are 4 GB de RAM. Algoritmii de tip Hidden Vector Entryption [59] și algoritmul bazat pe Inner Product au fost scrise complet în C++11, folosind bibliotecile PBC [94] și GMP [95], în timp ce implementarea HXT a fost preluată din [96]. Acesta folosește limbajul de programare Scala și Hadoop.

5.1.1.1 Experimente cu solutii bazate pe chei simetrice

Am rulat mai multe teste, pe trei baze de date, una de 1000 de intrări, una de 2000 de intrări și ultima pe 4000 de intrări. Am folosit aceeași bază de date pentru ambele soluții și multe intervale de adrese IP, de la subrețele la adrese minime și maxime aleatorii.

Prima parte a ambilor algoritmi constă în configurarea cheii și criptarea bazei de date. Acești pași se fac o singură dată, la început, apoi singura operațiune care rulează pe baza de date este decriptarea. Am derulat experimentele, în primul rând, pe o bază de date de 1000 de intrări. Pentru HXT, algoritmul de pornire constă din următorii pași: configurare inițială (34s), generarea pairing-urilor (333ms), generarea xtag-urilor (173ms), criptarea bazei de date (30ms) și generarea indexului HXT (68s). Pentru soluția bazată pe Inner-Product (IP), fazele de configurare și criptare au durat 102 secunde. Ulterior, am folosit o bază de date care conține 2000 de intrări. Configurarea inițială a durat 212 secunde pentru soluțiile bazate pe IP. În cazul HXT, xtag-urile, generarea indecșilor și criptarea bazei de date durează de două ori mai mult, proporțional cu dimensiunea bazei de date, în timp ce configurarea inițială și generarea asocierii rămân neafectate de modificare. În cazul unei baze de date cu 4000 de intrări, am măsurat o fază de configurare de 431 de secunde pentru soluția bazată pe IP și de aproximativ 300 de secunde pentru algoritmul HXT.

Am efectuat mai multe experimente când am încercat să extragem diferite intervale de IP-uri din baza de date. Am fost interesați să stabilim modul în care recuperarea unui anumit procent din baza de date afectează timpul total de interogare. Pentru fiecare dimensiune a bazei de date, am decriptat 1% din totalul intrărilor, apoi 3% și 5%. Am efectuat mai multe experimente pentru fiecare caz de utilizare, iar timpii de decriptare minim și maxim pentru fiecare experiment sunt furnizați în tabelele 5.1, 5.2 și 5.3.

Procentul de date decriptate	1%		3%		5%	
Soluție bazată pe IP (s)	62.39	55.6	62	91.8	125.86	103.59
Soluție bazată pe HXT (s)	18.34	18.38	34	18	34	18

Tabel 5.1 Rezultatele în secunde pentru o bază de date cu 1000 de intrări

Procentul de date decriptate	1%		3%		5%	
Soluție bazată pe IP (s)	77.6	24	62.8	83.5	96.6	92.6
Soluție bazată pe HXT (s)	35	36	36	34	70	37

Tabel 5.2 Rezultatele în secunde pentru o bază de date cu 2000 de intrări

Procentul de date decriptate	1%		3%		5%	
Soluție bazată pe IP (s)	57.6	92.8	70.4	107.5	107.7	123.6
Soluție bazată pe HXT (s)	76	78	86	82	84	82

Tabel 5.3 Rezultatele în secunde pentru o bază de date cu 4000 de intrări

Diferențele dintre soluțiile bazate pe IP sunt determinate de gama de adrese IP pe care o căutăm. În cazul căutării unor rețele întregi, începând de la rețea până la adresa de broadcast, timpul de decriptare este mai mic. În cazul opus, în cazul adreselor IP alese aleatoriu, interogările pot fi de până la două ori mai lente. Aceste rezultate sunt așteptate, pe baza modului în care este

proiectat algoritmul. Am observat mai puțin zgomot în cazul HXT, unde interogările sunt rezolvate în perioade similare de timp, indiferent de intervalele de adrese IP.

5.1.1.2 Exerimente pe algoritmi de criptare asimetrică

În cazul Hidden Vector Encryption, comparăm implementarea „naivă", în care adresele IP sunt codificate ca elemente binareîn vector cu cea „optimizată", unde adresa IP este codificată ca element unic al vecorului. În abordarea primului caz, viteza operațiunilor nu depinde de condiția de match, ci de dimensiunea rețelei.

Pentru o rețea de clasă A se efectuează următoarele operații: setup (4.13s) și keygen (0.34s). Pentru o rețea de clasă B, avem următoarele rezultate: setup (3.97s) și keygen (0.63s). Pentru o rețea de clasă C, avem următoarele rezultate: setup (5.29s) și keygen (1.10s) Acestea se fac o singură dată. Operațiunile de criptare și match sunt executate pentru fiecare intrare din baza de date. Pentru clasa A, avem operația de criptare care rulează în 2,48 secunde și operația de potrivire în 0,46 secunde pentru fiecare element. Pentru o rețea de clasă B, avem următoarele rezultate: criptarea se face în 2,43 s și potrivirea se face în 0,89 s. Pentru o rețea de clasă C, avem următoarele rezultate: criptarea se face 2.37s și potrivirea în 1.31s.

În abordarea optimizată, viteza crește, deoarece tablourile conțin doar un element. Operațiunile globale efectuate o singură dată sunt următoarele, setare, care a durat 461 ms și generarea de token, care a durat 62 ms. Operațiunile pe intrare in baza de date sunt următoarele: criptare, făcută în 118 ms și interogare în 79 ms. Deoarece am calculat operațiunile pe fiecare intrare din bază în configurația experimentală, am simulat rezultatele pentru baze de date mai mari, așa cum este prezentat în Figura 5.3.

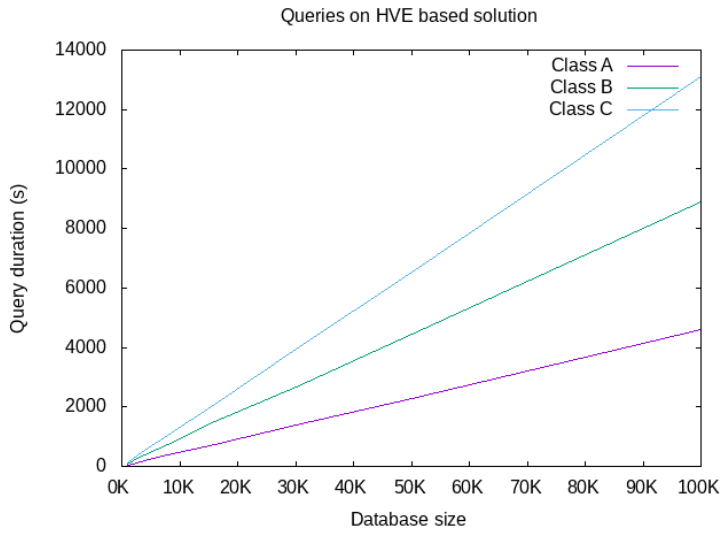

Fig.5.3. Simulări ale soluției HVE optimizate

Un avantaj pe care îl oferă această soluție este posibilitatea de a rula mai multe operațiuni în paralel. Nu există indexare a datelor și nu este necesar să se facă mai mulți pași secvențial. Am construit o versiune paralelă a algoritmului HVE folosind biblioteca C pthread din POSIX [97] pe o mașină virtuală cu patru procesoare. Am paralelizat algoritmul optimizat, unde vectorul conține un singur element. În acest caz, numărul maxim posibil de fire de execuție este patru și ele pot fi utilizate în funcția *setup*, unde algoritmul folosește *3*l+1* factori de orbire aleatorii, *l* fiind lungimea vectorului. Pentru această funcție, am obținut o îmbunătățire de 3,55 ori față de execuția cu un singur fir. La criptare, numărul maxim de fire de execuție posibile este trei, deoarece există doar trei variabile care pot fi calculate în paralel. În acest caz, îmbunătățirea este de 2,6 ori față de implementarea serială. Generarea de token-uri și interogarea pot fi paralelizate folosind două fire de execuție, în acest caz a unui singur vector element, ceea ce duce la o îmbunătățire de 1,7x. O comparație între cele două implementări seriale și paralele este prezentată în Figura 5.4.

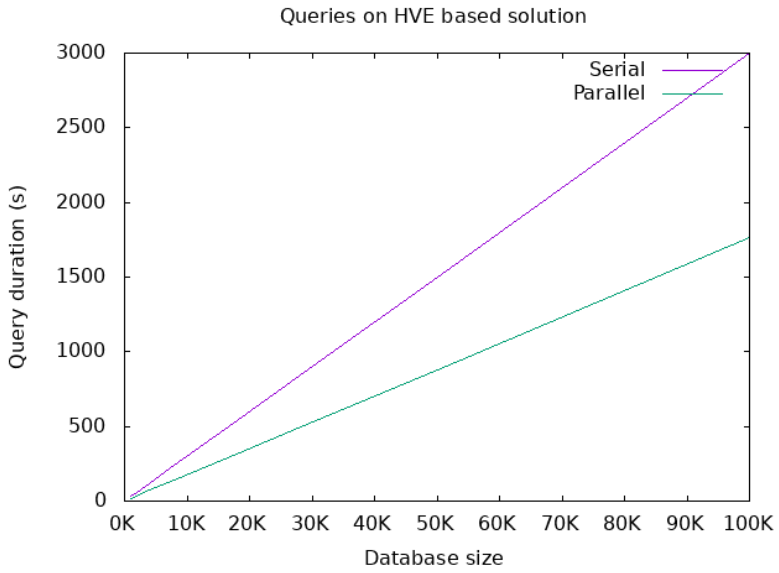

Fig.5.4.Simulare a soluției bazată pe HVE pentru sisteme seriale și paralele

Având în vedere experimentele de mai sus, observăm că abordările bazate pe chei simetrice rulează mai rapid, dar în aceste cazuri un singur utilizator poate cripta și decripta datele. Pe de altă parte, criptarea bazată pe HVE permite mai multor utilizatori să poată căuta prin datele criptate. Implementarea HVE oferă o relativă scalabilitate, deoarece există mai multe de inițializari de variabile și operațiuni care pot rula în paralel.

Criptarea simetrică permite interogări pe intervale de adrese IP, împreună cu căutarea exactă a unui port, în timp ce HVE permite setarea unei taxonomii de bază a porturilor și

examinarea lor folosind interogări de subseturi. În cazul criptării asimetrice, modul în care datele sunt codificate este important, ceea ce duce la rezultate mult mai bune decât abordarea „naivă”, în care adresa IP este transformată în vector binar.

5.2 Căutări private în date financiare

Acest proiect include implementarea mai multor algoritmi cunoscuți, inclusiv cei prezentați în secțiunea anterioară. Fiecare dintre sistemele implementate este testat pentru a înțelege capacitățile și limitările sale. Combinăm sistemele de criptare și codificarea datelor pentru a efectua interogări pe o bază de date care conține informații criptate despre tranzacțiile financiare. Pentru acest proiect, o tranzacție este definită de următoarele informații: IBAN-ul expeditorului, IBAN-ul destinatarului, ziua, luna și anul tranzacției și valoarea tranzacției. Pentru ultimul element vor fi efectuate diverse interogări.

În Fig. 5.5, există o diagramă care ilustrează un posibil scenariu de utilizare pentru efectuarea de interogări pe o bază de date care deține informații despre tranzacțiile financiare. Bob vrea să facă o plată către Alice. El anunță banca. Banca criptează tranzacția și o stochează pe un server. Acum, Alice vrea să-și verifice contul și să vadă dacă a primit plata de la Bob. Ea notifică banca despre intenția sa, iar banca trimite un token de căutare pe baza cererii ei și o cheie de decriptare. Alice trimite token-ul la serverul de stocare și primește înapoi o listă de înregistrări criptate care se potrivesc cu token-ul. Ea decriptează înregistrările folosind cheia de decriptare emisă de bancă.

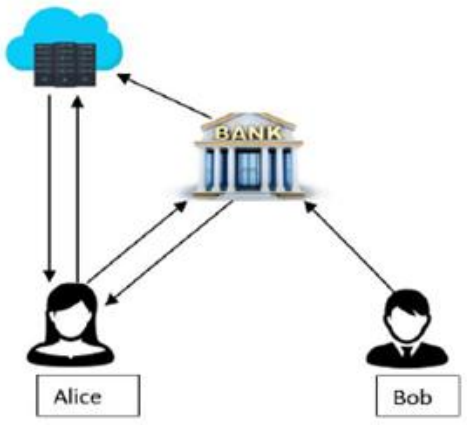

Fig. 5.5 Scenariul de utilizare pentru interogarea bazei de date criptate

5.3 Concluziile capitolului

În acest capitol, am analizat în detaliu o soluție orientată spre cercetare, numită *searchable encryption*. În primul rând, am extins o schemă deja existentă, am adăugat partea de decriptare a datelor și am comparat-o cu un sistem bazat pe „index inversat”. Sistemul nostru a arătat rezultate mai bune în cazul intervalelor care acoperă un procent mai mare din baza de date totală, dar timpul de căutare crește substanțial în cazul intevalelor aleatoare de adrese IP.

Criptarea simetrică permite interogări de interval în cazul adreselor IP împreună cu căutarea exactă a unui număr de port, în timp ce HVE permite setarea unei taxonomii de bază a porturilor și examinarea lor folosind interogări de subseturi. În cazul criptării asimetrice, formatele adecvate de codificare a datelor conduc la algoritmi mai rapizi.

Apoi am efectuat experimente pe date financiare și am obținut rezultate promițătoare, inclusiv decriptarea unei baze de date mici cu 100 de înregistrări în aproximativ 0,3 secunde. Cu toate acestea, sunt necesare îmbunătățiri suplimentare pentru ca tehnicile de căutare în date criptare să devină potrivite pentru aplicațiile din viața reală.

6 PROCESAREA DATELOR

Procesarea textului este una dintre cele mai comune aplicații ale învățării automate în prezent. În acest capitol, prezentăm modalități de clasificare a textului folosind un model personalizat sau un algoritm de ultimă generație și analizăm costul general pe care confidențialitatea o aduce algoritmilor. Scopul principal al acestui capitol este de a înțelege dacă confidențialitatea poate fi implementată în algoritmii de învățare automată, care sunt condițiile adecvate și limitările acesteia. În cea de-a doua parte a capitolului, prezentăm un algoritm de clasificare a textelor care rulează scurte texte tehnice. Scopul său aici este de a spori nevoia de confidențialitate, demonstrând că informațiile pot fi obținute din orice tip de date.

6.1 Clasificarea textelor folosind Federated Learning

În această secțiune, am folosit un algoritm actual pentru a rula clasificarea textului și apoi am adăugat confidențialitate schemei. Am efectuat câteva teste pentru a determina diferențele în timpii de procesare pentru texte de diferite lungimi și am ajuns la câteva concluzii din rezultatele obținute.

În primul rând, am pregătit setul nostru de date care conține articole științifice referitoare la domeniile energiei solare și medicamentelor. Apoi, am început un proces de curățare pentru a obține o bază de date validă. Am eliminat cuvintele non-alfabetice, precum și cuvintele de legătură. Pasul final a fost lematizarea, folosită pentru a converti fiecare cuvânt în forma lui de dicționar. Pentru a rula algoritmul, am codificat cuvintele ca numere întregi.

In ceea ce priveste modelul folosit, am ales un model GRU simplu cu un singur strat cu functie de activare sigmoida. Am propus o arhitectură cu două instanțe separate, numițe sugestiv Alice și Bob, cu propriile lor date private, așa cum se arată în Figura 6.1. Din baza noastră de date, am trimis jumătate din date la ambele mașini. Aveam un nod central (sau un server de învățare federat) care a servit ca agregator pentru modelele actualizate. Prin urmare, atunci când modelul nostru a fost trimis atât către Alice, cât și către Bob, modelul a fost actualizat continuu și personalizat cu propriile date, apoi trimis înapoi la componenta centrală, unde modelul complet a fost actualizat corespunzător.

Pentru antrenare și evaluare, am trunchiat datele noastre la 5000, 2000 și 1000 de cuvinte per articol. Atât la abordările centrale, cât și la cele securizate, am obținut o precizie similară. Singura diferență a fost în timp, modelul distribuit fiind de 1,5x, 3x, respectiv de 4,5x mai lung, dar cu avantajul confidențialității. Pentru testare, am folosit framework-ul PySyft [71].

În continuare, am vrut să ne testăm ipoteza că, cu mai puține date pentru fiecare mașină, diferența dintre modelul central și cel federal ar crește în timp. Deci, am folosit datele de colectare a mesajelor spam prin SMS și am trunchiat fiecare tweet la 30 de cuvinte. Modelul

Federated, care asigură confidențialitatea, a fost între 15 și 18 ori mai lent decât cel central. Aceasta poate părea o scădere a performanței ridicate, dar am folosit doar doi clienți și fiecare dintre ei avea foarte puține date de procesat. În acest caz, partea cea mai costisitoare a fost în principal trimiterea și primirea actualizărilor modelului.

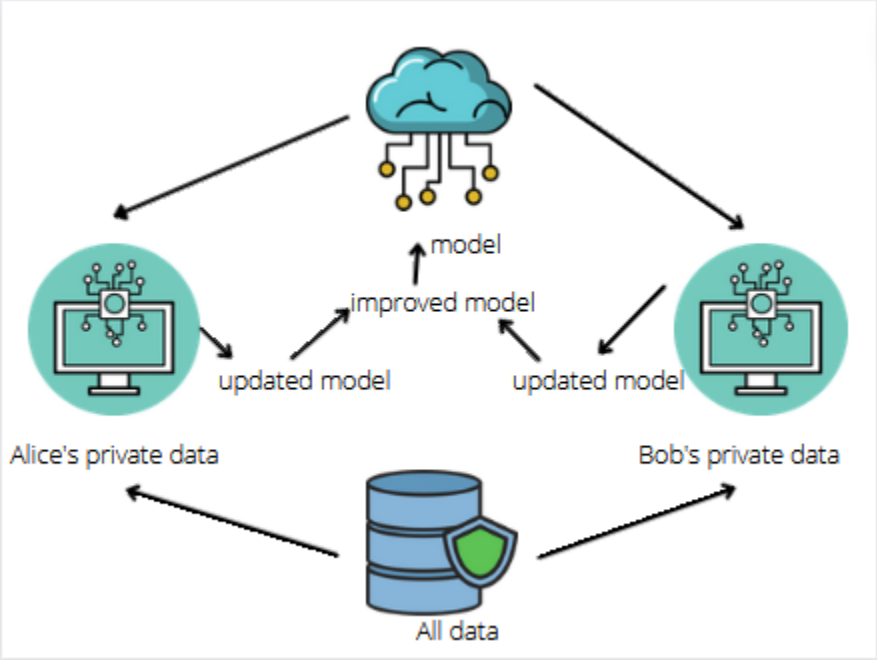

Fig. 6.1 Soluția propusă pentru modelul de Federated Learning

Pentru clasificarea articolelor, am trunchiat lungimea unui articol la 5000, 2000 și 1000 de cuvinte. Am făcut asta pentru că dorim o comparație mai bună a comportamentului algoritmului. Toate testele au fost efectuate folosind Google Colaboratory1. Am ales acest lucru datorită simplității și specificațiilor hardware bune:

• 2vCPU @2.2GHz

• 13GB RAM

• 100GB Free Space

6.1.1.1 Medicamente versus energie solară

Pentru 5000 de cuvinte per articol, putem vedea în Tabelul 6.1 că timpul de rulare pentru soluția Federated este de 1,35 ori mai lung față de PyTorch [79]. Aceasta nu este mult având în vedere caracteristicile suplimentare pe care le-am adăugat. Pentru modelul centralizat, am rulat pentru 4 epoci, în timp ce pentru abordarea Federată doar 2, pentru că dura prea mult. După

cum se poate observa, avem o precizie destul de apropiată. Foarte probabil, dacă am fi setat în ambele cazuri 4 epoci, acuratețea pentru modelul Federated ar fi fost foarte apropiată de cea centralizată.

Tabel 6.1 Comparația dintre cele 2 modele pentru articole de 5000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	4	2
Acuratețe	78.83	72.73
Timpul total	7h27min	5h24min
Timul per epocă	1h52min	2h42min
Rămân datele private?	NU	DA

Pentru 2000 de cuvinte pe articol, am obținut rezultate similare în ceea ce privește acuratețea modelului. Am antrenat atât modelul central, cât și pentru model distribuit timp de 10 epoci. Am observat însă că overhead-ul în timp creștea și abordarea distribuită a durat de patru ori mai mult de această dată. Rezultatele comparative sunt expuse în Tabelul 6.2

Tabelul 6.2 – Comparația dintre cele 2 modele pentru articole de 2000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	10	10
Acuratețe	78.5	78.11
Timpul total	1h57min	6h
Timul per epocă	12min	36min
Rămân datele private?	NU	DA

Pentru 1000 de cuvinte pe articol, obținem o acuratețe aproape identică. Antrenăm ambele modele timp de 15 epoci. După cum era de așteptat, modelul Federated rulează mai greu, de data aceasta de 4,5 ori mai mult decât abordarea normală.

Tabelul 6.3 – Comparația dintre cele 2 modele pentru articole de 1000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	15	15
Acuratețe	72.78	72.79
Timpul total	1h8min	4h31min
Timul per epocă	4min39sec	18min
Rămân datele private?	NU	DA

6.1.1.2 Energie fotovoltaică versus regenerabilă

În această secțiune, obținem rezultatele pentru două domenii similare: energie fotovoltaică și energie regenerabilă. După cum ne așteptam, am obținut rezultate mai proaste decât în runda anterioară, unde am avut mai multe subiecte care nu au legătură între ele. Am testat în același mod, pentru articole de 5000, 2000 și 1000 de cuvinte.

Pentru testul de 5000 de cuvinte, am obținut o precizie de aproximativ 65%. Ca și în rularea anterioară, abordarea Federated necesită de 1,5 ori mai mult timp. Pentru lungimea articolelor de 2000 am rulat 10 epoci. Precizia este de aproximativ 61% și modelul Federated durează de 2,5 ori mai mult decât cel central, similar cu ceea ce am obținut în prima secțiune. Pentru o lungime de 1000 de cuvinte, așa cum ne așteptam, abordarea Federated necesită de aproximativ de 4,5 ori mai mult timp față de Pytorch. Rezultatele se găsesc în Tabelele 6.4, 6.5 și 6.6.

Tabel 6.4 Comparația dintre cele 2 modele pentru articole de 5000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	4	2
Acuratețe	59.9	64.59
Timpul total	6h12min	4h36min
Timul per epocă	1h33min	2h18min
Rămân datele private?	NU	DA

Tabel 6.5 Comparația dintre cele 2 modele pentru articole de 2000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	10	10
Acuratețe	60.69	57.85
Timpul total	3h16min	7h51min
Timul per epocă	19min	47min
Rămân datele private?	NU	DA

Tabel 6.6 Comparația dintre cele 2 modele pentru articole de 1000 de cuvinte

	Model centralizat	**Model distribuit**
Numărul de epoci	15	15
Acuratețe	61.71	57.5
Timpul total	1h13min	4h14min
Timul per epocă	4min48sec	17min
Rămân datele private?	NU	DA

6.1.1.3 Setul de date SMS Spam

În această secțiune, am testat modelul nostru pe setul de date de colectare a mesajelor spam prin SMS. Am trunchiat lungimea fiecărui tweet la 30 de cuvinte pentru a ne putea antrena pentru 100 de epoci. Precizia este mare, așa cum era de așteptat, din cauza numărului de epoci

pe care le-am antrenat. Ceea ce este interesant este diferența de timp dintre cele două abordări. După cum am observat până acum, mai puține cuvinte pe articol, mai mult timp modelului federat, ceea ce se confirmă și aici. Pentru tweet-urile de 30 de cuvinte, modelul distribuit ia nu mai puțin de 12 ori mai mult timp decât cel central. Acest comportament poate fi justificat de numărul foarte mic de mașini distribuite (am folosit doar două, Bob și Alice). Rezultatele pentru 100 de epoci pot fi văzute în Tabelul 6.7 de mai jos.

Table 6.7 Comparația celor 2 modele pentru texte scurte

	Model centralizat	**Model distribuit**
Numărul de epoci	100	100
Acuratețe	98.04	96.17
Timpul total	8min	1h31min
Timul per epocă	4.8s	55s
Rămân datele private?	NU	DA

Comunicarea dintre serverul central și cele de procesare este un blocaj clasic în Federated Learning. Această problemă afectează capacitatea maximă a unui model care poate fi trimis și acuratețea acestuia. Atenuarea acestor probleme ar putea fi reducerea arhitecturii modelului sau eliminarea unor parametri. Acest lucru va duce la o precizie mai slabă, dar experiența utilizatorului va fi îmbunătățită. O altă abordare ar fi antrenarea unor subseturi ale modelului global pe fiecare dispozitiv, o metodă numită Federated Dropout, detaliată în [99].

6.2 Clasificarea textelor tehnice, scurte

În acest capitol, analizăm procesarea textului în cazurile în care algoritmul de învățare automată nu reușește să returneze rezultate optime.

Scopul GEO Knowledge Hub este de a oferi, într-un singur loc, acces la toate documentele asociate aplicațiilor de observare a Pământului. Aceste documente includ lucrări de cercetare, rapoartele care descriu metode de lucru și rezultatele algoritmilor software utilizați pentru prelucrarea datelor.

Principala problemă legată de aceste tipuri de baze de date este numărul mare de informații nestructurate din interior. Orice căutare va ridica un număr mare de rezultate, ceea ce poate fi confuz pentru un utilizator. O mulțime de informații esențiale se pot pierde după primele pagini și multe seturi de date conectate se pot pierde din cauza unui număr mare de rezultate preluate. În ciuda cunoștințelor pe care le oferă o bază de date complexă, dificultatea de a prelua date specifice poate depăși beneficiile.

Scopul implementării este de a construi un algoritm de clasificare pentru texte scurte, inclusiv seturi de date, care conțin câteva cuvinte tehnice proeminente și sunt greu de clasificat de soluțiile actuale.

Scopul principal al proiectului este de a construi un sistem de căutare „inteligent”, care să fie capabil să asocieze lucrări științifice cu servicii web, să obțină date din surse multiple într-un mod coerent și să arate rezultatele în funcție de relevanța lor. Pentru utilizatori, acest proces este transparent, ei încep prin a căuta în serverul web folosind cuvinte cheie sau text liber și, în spate, există un clasificator care oferă nu doar rezultatele obținute prin interogare, ci și intrări legate de subiect, utile unui expert. Figura 6.2 explică imaginea de ansamblu a procesului.

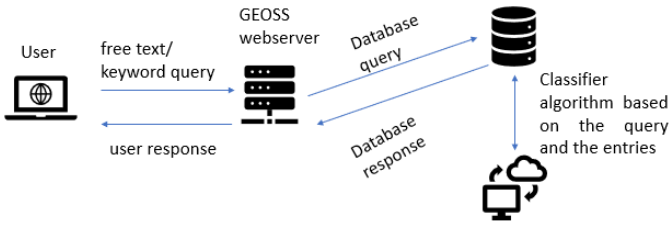

Fig.6.2 Algoritmul de clasificare integrat în sistemul complet

În cazul textelor scurte, cum ar fi înregistrările de metadate, cea mai bună abordare este de a construi o ierarhie de cuvinte predefinite legate de subiect și de a atribui automat fiecare text acestor categorii. În acest caz, este nevoie de o viziune de specialitate pentru a oferi vocabularul și ierarhia necesare. În acest caz arborele a fost construit de la informații generale la mai specifice legate de energie, începând cu cel mai general nivel, există „Temă” → „Domeniu” → „Subdomeniu” → „Informații” → „Măsuri”.At the “Theme” level, we have “Energy”.

- La nivel de „Domeniu”, există „Energie regenerabilă”.
- La nivelul „Subdomeniu”, există „Energie solară”.
- La nivelul „Informații” sunt „Resurse solare”, „Atmosferă și meteorologie”, „Topografia solului”, „Anul meteorologic” și „Potențial solar”, împreună cu termenii cei mai proeminenți ai fiecărei categorii.
- Fiecare intrare de tip „Măsuri” este legată de fiecare categorie de mai sus și oferă informații suplimentare despre vocabularul tehnic pentru fiecare caz. Cuvinte precum „eșantionare în timp”, „rezoluție în timp”, „model numeric de predicție a vremii” sunt considerate relevante pentru acest nivel.

6.2.1 Observații

După rularea algoritmului de clasificare, informația a fost împărțită în cinci câmpuri, urmărind subiectele principale de la nivelul „Informații" . Fiecare text a fost atașat cu un scor de similaritate de un subiect. Primele comparații pe care le-am făcut sunt relative la căutările manuale.

Primul domeniu tehnic este „energia solară", iar primele zece rezultate obținute sunt legate de expunerea solară globală, atlasul despre potențialului solar în țările dezvoltate de institutele de cercetare, modelul computerizat de radiații al Biroului de Meteorologie (BOM) [111] privind expunerea solară și diferite hărți de iradiere împărțite pe regiune. În comparație, căutarea manuală a aceluiași concept returnează rezultate legate de măsurătorile de la sol pentru stațiile din Rețeaua de monitorizare și cartografiere a resurselor regenerabile (RRMM) Solar Radiation Monitoring Network. Aceste tipuri de conținut sunt similare, iar rezultatele pot fi grupate automat în categorii. Soluția noastră oferă 285 de rezultate precise, în timp ce căutarea manuală 790.

Al doilea domeniu tehnic este „Atmosferă și meteorologie". În acest caz, cele mai bune potriviri sunt legate de seturile de date privind intensitatea vântului, temperatura geotermală și stațiile meteorologice. De asemenea, au fost găsite unele rezultate care nu conțin cuvintele cheie în titlu, dar oferă o intrare interesantă în rezumat. De exemplu, algoritmul a cartografiat măsurătorile de la sol pentru stațiile din Rețeaua de monitorizare și cartografiere a resurselor regenerabile (RRMM) Solar Radiation Monitoring Network. Acestea oferă date de rezoluție de un minut pentru trei componente solare, împreună cu temperatura, umiditatea, viteza și direcția vântului și presiunea barometrică. Chiar dacă titlul acestor resurse este mai degrabă legat de componentele solare, conținutul oferă și date importante despre meteorologie. Aici, o căutare prin cuvinte cheie va oferi mult mai puține rezultate. Algoritmul oferă 165 de potriviri exacte pe subiect. În cazul căutării manuale, găsim un singur rezultat dacă căutăm ambii termeni în același timp.

Al treilea subiect principal este „topografia solului". Algoritmul obține 165 de rezultate potrivite, în timp ce căutarea manuală nu oferă niciun rezultat în baza de date. Căutarea unor termeni separați duce la 510 rezultate. Rezultatele de top ale clasificării automate conțin seturi de date despre liniile de cale ferată și rutieră, puncte de aeroport, linii de transmisie, zone protejate, granițe administrative ale țării și multipoligoane de cotă. Aceste cuvinte se potrivesc cu etichetele predefinite ale expertului. Căutarea manuală pentru acest subiect consideră hărțile vitezei vântului drept cele mai relevante rezultate.

Al patrulea domeniu este „anul meteorologic". Algoritmul returnează hărți de iradiere orară, zilnică, lunară și anuală, mediile iradierii solare globale zilnice, resursele solare medii lunare și hărțile anuale ale vântului. Căutarea manuală returnează ca la început intrări similare, seturi

de date care conțin radiația solară medie anuală (DNI, GHI, BHI, DHI), hărți de iradiere anuală, dar și valori zilnice ale iradierii solare de suprafață, care pot fi mai puțin conectate la subiect. Un rezultat interesant pe care l-am găsit cu ușurință, dar considerat mai puțin relevant de căutarea manuală, este un atlas al vântului, adică viteza medie anuală a vântului și hărți specifice de producție la patru niveluri (25, 50, 75 și 100 m) deasupra solului și mării. De asemenea, am găsit o hartă globală a presiunii atmosferice medii lunare și anuale pe 22 de ani furnizată de NASA, în timp ce căutarea manuală a găsit hărți speranța de pierdere a vieții pentru diferite sectoare de activitate.

Al cincilea câmp este „potențialul solar”. Rezultatele de top sunt similare pentru căutarea automată și manuală, legate de potențialul tehnic și teoretic de producere a energiei solare. Următoarele rezultate oferite de căutarea manuală sunt legate de iradierea solară, la fel ca cele din etichetele anterioare, în timp ce sistemul de clasificare găsește seturi de date despre producția de energie electrică fotovoltaică și zone de oportunitate ale proiectelor de concentrare a energiei solare.

6.2.2 Concluziile capitolului

Analizând rezultatele de mai sus, cea mai bună abordare legată de gruparea înregistrărilor despre observațiilor solare este o abordare hibridă care include o combinație de etichete manuale și tehnici de învățare automată. În acest caz, este necesară o listă de etichete predefinite, deoarece temele, domeniile sau informațiile solicitate sunt greu de descoperit automat. Principala provocare a clasificării intrărilor este legată de lungimea textelor și de conceptele tehnice. Este foarte important de știut în prealabil ce este relevant din perspectiva unui expert, iar algoritmul de învățare automată poate oferi o nouă perspectivă asupra asemănărilor de text și a rezultatelor interesante. Putem obține intrări din baza de date care nu sunt returnate automat la interogare. Putem folosi un sistem de clasare, bazat pe textul de căutare, astfel încât rezultatele cele mai relevante vor fi afișate mai întâi, pentru o experiență mai bună a utilizatorului.

7 CONCLUZII

Lucrarea din această teză explorează mai multe soluții pentru confidențialitate la fiecare nivel de prelucrare a datelor. Deoarece big data și confidențialitatea sunt termeni complecși și generali care duc la oportunități largi de cercetare, este imposibil să se construiască niveluri sigure în mod general, care să acopere mai multe scenarii și tipuri de date.

Prezenta teză se concentrează pe îmbunătățirea confidențialității în big data, parcurgând în același timp o listă de pași de procesare redefiniti: colectarea, colectarea și transmiterea, stocarea și procesarea datelor. Lucrarea din această teză acoperă mai multe domenii, de la securitate și confidențialitate până la procesarea limbajului natural. Conține contribuții originale în domeniul criptografiei care păstrează confidențialitatea, obscurizării locației, securității în IoT, educației deschise și procesării limbajului natural, permițând cercetări suplimentare în oricare dintre aceste domenii.

Toate contribuțiile au fost publicate în [9], [10], [11], [12], [13], [14], [15] și [16]. Pentru a putea lucra la teză a fost nevoie de abilități tehnice diferite, de la utilizarea mai multor limbaje de programare până la înţelegerea structurilor din algebra, cum ar fi grupuri, inele și câmpuri. Interconectarea tuturor acestor domenii contribuie la dezvoltarea soluțiilor si cunoștințelor actuale, permițând astfel dezvoltarea unor soluții mai eficiente orientate spre confidențialitate.

Toate contribuțiile pot fi împărțite după cum urmează:

Contribuția 1: Principalele protocoale în IoT și soluțiile lor de securitate

Articole:

- *Survey of standardized protocols for the Internet of Things* [9]
- *Challenges in security in Internet of Things* [10]

Această contribuție este mai degrabă o explorare a stivei de rețea a dispozitivelor embedded, a principalelor protocoale și a soluțiilor de securitate adaptate. În general, dispozitivele IoT au memorie scăzută, capacitate redusă a bateriei, capacități de procesare reduse și condiții radio vulnerabile. Stiva standard TCP/IP nu este potrivită pentru acest mediu, așa că grupurile de lucru au început să dezvolte protocoalele existente la versiuni noi. A fost luată în considerare o schemă de adresare, cum ar fi IPv6, deoarece există miliarde de noduri interconectate. Mai multe grupuri de lucru au început deja să standardizeze protocoalele specifice IoT, cum ar fi 6LoWPAN (RFC 4944 și RFC 6282), IEEE802.15.4 și ZigBee și să descrie modalități de activare a IPv6 în medii constrânse. Alte cerințe găsite se referă la securitate și confidențialitate, deoarece numărul atacurilor de tip Denial of Service a crescut recent. La nivelul Aplicație, o modalitate obișnuită de a prelua și solicita date este utilizarea arhitecturii Web și, mai precis, HTTP. Aceasta utilizează URI-uri ca identificatori de resurse și se bazează pe arhitectura REST pentru a publica informații.

Pentru dispozitivele embedded, există un grup de lucru IETF, Constrained RESTful Environments (CoRE), care își propune să dezvolte protocoale RESTful, compatibile cu HTTP pentru dispozitivele cu resurse limitate. Cel mai proeminent protocol de acest gen este CoAP, la nivelul Aplicație. Am prezentat un studiu al celor mai utilizate protocoale standardizate pentru Internet of Things. Investigăm protocoalele de nivel de Aplicație (CoAP, MQTT), protocoalele de descoperire a serviciilor (mDNS, DNS-SD, uBonjour) și protocoalele de infrastructură (IEEE 802.15.4, 6LoWPAN, LoRaWAN). Am prezentat diferite moduri de integrare a protocoalelor la nivelul Aplicație, cum ar fi MQTT, CoAP și HTTP. În cele din urmă, propunem un sistem de casă inteligentă cu noduri senzori wireless și asistenți robot care utilizează protocoale IoT standardizate (IEEE 802.15.4, 6LoWPAN, CoAP).

Am extins munca de la primul articol cu o privire de ansamblu asupra securității acestor protocoale. Am ajuns la concluzia că standardizarea stivei de rețea și a securității este un factor cheie pentru succesul IoT. O stivă de rețea pentru sistemele IoT poate fi creată folosind, la fiecare nivel, protocoale dezvoltate special pentru medii constrânse. La nivel Fizic și MAC, există protocoale bazate pe 802.15.4, care pot fi găsite pe mai multe dispozitive. La nivelul Rețea putem folosi un protocol care oferă rutare bazată pe adrese MAC. Un nou strat de adaptare este adăugat la stiva dintre, între Rețea și Transport, utilizat pentru furnizarea de adrese IPv6 dispozitivelor IoT. La nivelul Aplicație, protocoalele au fost adaptate pentru a transporta mai des date mai puține. A învăța cum să se dezvolte un sistem inteligent pentru clădiri, campusuri și case duce la dobândirea de cunoștințe atât despre avantajele, cât și despre problemele legate de conectarea dispozitivelor multiple și eterogene. Pe de o parte, acestea pot îmbunătăți calitatea vieții și utilizarea resurselor, dar vin și cu riscuri de securitate și confidențialitate. Adăugarea securității la IoT este o sarcină dificilă, deoarece cele mai sigure mecanisme din stiva TCP/IP nu sunt potrivite. Protocolul standard DTLS poate fi utilizat, dar împreună cu îmbunătățiri care duc la mai penalizări de performanță diminuate. De asemenea, pot fi dezvoltate noi protocoale de securitate, atunci când se încearcă minimizarea handshake-ului inițial.

Am propus o soluție de securitate pentru stiva de rețea de mai sus, bazată pe chei pre-partajate, împreună cu un algoritm existent. Am decis să folosim derivarea cheilor cu AES-128, deoarece este suportat hardware pe multe dispozitive IoT și să adăugăm noi opțiuni „AUTH” și „AUTH_MSG_TYPE” pentru mesajele dintre server și client. Fiecare client are o cheie configurată și serverul are un set toate id-urile posibile. Primul mesaj trimis de la client către server conține id-ul clientului și un token unic. Serverul găsește parola asociată dispozitivului, derivă cheia, nonce_1 și trimite un mesaj criptat cu rol de challenge. Când clientul primește challenge-ul, folosește cheia partajată pentru a decripta pachetul, primește cheia și nonce_1 și trimite un răspuns folosind cheia derivată și token-ul. Dacă token-ul este același ca în transmisia inițială, atunci clientul este autentificat. Criptăm payload-ul de la nivelul Aplicație și utilizăm opțiunile din antetul CoAP pentru a separa interogările criptate de cele cu text clar.

Problema principală este că cheile pre-partajate pot fi sparte, astfel încât propunerea noastră de sistem explorează posibile atacuri, prin analizarea tiparelor de trafic pe partea de server. Dacă dispozitivele rău intenționate au fost autentificate, utilizatorii trebuie să fie alertați și sistemul ar trebui configurat cu o altă soluție, cum ar fi cheile asimetrice.

Contribuţia 2: Ascunderea locației pentru utilizatori mobili

Articole:

- *Teaching privacy through the development and testing of a location obfuscation solution* [11]
- *Location privacy for non-stationary users* [12]

Necesitatea menținerii confidențialității locației a devenit mai puternică în mediul actual, când serviciile ce folosesc locația sunt peste tot. Sistemele actuale sunt capabile să monitorizeze pe oricine în timp real, cu o precizie incredibilă. Împreună cu progresele tehnologice în puterea și capacitatea de procesare, populația poate fi urmărită oriunde ar merge, mult mai ușor decât oricând. De exemplu, locația cuiva poate fi folosită în mod rău intenționat pentru a trimite mesaje spam care conțin produse sau servicii legate de locația victimei. Un alt risc dat de neglijarea confidențialității locației este dat de abuzul fizic. Un atacator poate folosi locația cuiva pentru a urmări, jefui, ataca această victimă. Un al treilea motiv pentru care lipsa confidențialității locației poate cauza prejudicii unei persoane este că locația poate spune lucruri despre o persoană. Doar știind unde a fost cineva în perioada trecută poate duce la presupuneri corecte despre religie, orientare politică, starea de sănătate și alte informații private similare.

Am pornit de la un algoritm de obscurizare de bază, care folosește distribuția Planar Laplace pentru a adăuga zgomot la o locație reală și am demonstrat că nu este suficient de fiabil. Din punct de vedere al securității, triangularea este o amenințare pentru locațiile obscurizate cu algoritmul de bază. În plus, unele locații generate în acest fel pot să nu fie plauzibile, deoarece pot fi plasate fie în zone ciudate, fie în zone normale, cu aceeași probabilitate. Atât aceste defecte, cât și orice altele legate de acestea au fost abordate prin dezvoltarea unei noi versiuni a algoritmului. Rezultatele, prezentate ca o comparație între soluția noastră și punctul de plecare de ultimă generație, evidențiază o îmbunătățire semnificativă. Fiecare locație generată cu algoritmul nostru aparține unei zone relativ populare. Singurul criteriu pentru generarea locațiilor este scorul de popularitate. Distanța dintre locația reală și cea fabricată nu mai este relevantă pentru algoritm. Distanța maximă este limitată de nivelul de obscurizare specificat ca intrare, dar nu este impusă în niciun alt mod.

Am dezvoltat algoritmul pentru a crea o cale credibilă pentru utilizatorii de telefonie mobilă. Am generat noi locații pe baza scorului de popularitate al punctelor de interes din apropierea utilizatorului și pe locația anterioară a acestuia. Rezultatele sunt în concordanță cu

așteptările teoretice. Având în vedere o traiectorie bazată pe locații reale, rezultatul pare foarte asemănător ca orientare, cu locații generate în locuri plauzibile.

După construirea algoritmului, l-am testat împotriva atacurilor de ultimă generație asupra mecanismelor de obscurizare a locației. Algoritmul se comportă similar în cele mai multe cazuri, dar există câteva scenarii marginale care trebuie luate în considerare. Atunci când locația obscurizată, este cumva plasată într-un loc nepopular de pe hartă, algoritmul nu va fi niciodată spart de atacul cu forță brută. Când zona locației obscurizate este puțin populară și unele zone mai populare sunt în apropiere, algoritmul este probabil să aleagă acele locuri populare ca rezultat. Acest lucru îl poate determina pe atacator să creadă că zona care conține locația reală este mai mică decât este în realitate. Cu toate acestea, scenariul obișnuit se potrivește cu comportamentul algoritmului original, ceea ce înseamnă că soluția noastră oferă și protecție împotriva atacului cu forță brută.

Contribuția 3: Stocarea securizată a datelor

Articole:

- *Practical analysis of searchable encryption strategies for financial architecture* [13]
- *Sharing of Network Flow Data across Organizations using Searchable Encryption* [14]

Analiza tiparelor de trafic poate ajuta organizațiile să aloce resurse mai bine în ceea ce privește alocarea timpului angajaților și achiziția de echipamente și poate ajuta, de asemenea, la îmbunătățirea colaborărilor inter-organizaționale. Prin urmare, partajarea datelor privind fluxurile de circulație a datelor în rețea poate îmbunătăți semnificativ optimizările operațiunilor. Ca un aspect conex, unele sarcini inter-organizații, de exemplu, răspunsul la incidente de securitate cibernetică, necesită ca mai multe organizații să-și investigheze în comun jurnalele de rețea pentru a înțelege natura și amploarea amenințării. Pe de altă parte, atunci când aveți de-a face cu jurnalele de rețea, securitatea și confidențialitatea devin preocupări importante. Dezvăluirea jurnalelor de trafic de rețea ale unei organizații poate permite unui adversar să afle detalii despre interese comerciale confidențiale. Pentru a evita astfel de riscuri de confidențialitate, am avut în vedere searchable encryption, un instrument puternic care poate fi folosit pentru a proteja datele. În loc să-și partajeze datele în text clar, fiecare organizație își criptează mai întâi setul de date cu un tip special de criptare care permite căutarea direct pe texte criptate. Apoi, pe baza unor scenarii specifice, o organizație poate acorda unei alte părți capacitatea de a căuta și de a prelua unele dintre datele sale pe baza unor obiecte criptografice numite token-uri de căutare. Folosind un token, se poate efectua căutarea peste date criptate, fără a fi nevoie de decriptare. Căutarea se efectuează pe baza unui predicat. De exemplu, accesul poate fi acordat numai pentru pachetele care provin de la o anumită adresă IP sau la un anumit

port de destinație. Ca rezultat al căutării, sunt obținute numai texte clare care se potrivesc cu predicatul de căutare. Elementele de date care nu se potrivesc rămân criptate și nicio informație despre acestea nu este dezvăluită în timpul căutării. Am implementat interogări complexe, care includ diverse combinații de IP și port de destinație, de exemplu, trafic HTTP către un server sau trafic SSH către un punct de intrare (entry point).

Am investigat utilizarea mai multor tehnici bine cunoscute de căutare prin date criptate pe scenariul specific al inspectării jurnalelor de rețea criptate în organizații. La nivelul datelor de rețea ne-am concentrat pe granularitatea fluxurilor, care conține atribute sub formă de metadate care sunt relevante în majoritatea scenariilor de utilizare (de exemplu, adrese IP, numere de porturi, numărul de octeți și pachete transferate per conexiune, tip de protocol etc.). Principala provocare care apare atunci când se utilizează searchable encryption este dată de costul suplimentar de resurse de procesare pe care îl aduce. Când se aplică scheme de ultimă generație direct asupra traficului de rețea, interogarea datelor necesită mult timp. Am găsit niște encodări adecvate și am propus o abordare personalizată pentru traficul de rețea la nivel de granularitate a fluxurilor pentru a aduce overheadul de procesare la niveluri acceptabile.

Am descoperit că abordările cu chei simetrice rulează mai rapid, dar, în aceste cazuri, un singur utilizator poate cripta și decripta datele. Pe de altă parte, o abordare cu cheie asimetrică, cum ar fi hidden vector encryption (HVE) permite mai multor utilizatori să poată căuta peste texte criptate. Implementarea HVE oferă scalabilitate, deoarece există o mulțime de inițializari variabile și operațiuni care pot rula în paralel. Criptarea simetrică permite interogări peste intervale de adrese IP, împreună cu căutarea exactă a unui număr de port, în timp ce HVE permite setarea unei taxonomii de bază a porturilor și examinarea lor folosind interogări cu subseturi. În cazul criptării asimetrice, modul în care datele sunt encodate este important, ceea ce duce la rezultate mult mai bune decât abordarea „naivă”, în care adresa IP este transformată în vectori binari.

Ulterior, am implementat două scheme de Hidden Vector Encryption și două sisteme de suport cu inner product. Am efectuat o evaluare amănunțită a corectitudinii și performanțelor acestora pentru diverse scenarii de utilizare și pentru diferite seturi de parametri, folosind o bază de date care conține informații despre tranzacțiile financiare. Am ales mai multe encodări pentru valoarea tranzacției care acceptă interogări peste intervale de valori sau potriviri exacte sau ambele. Au fost obținute rezultate promițătoare, inclusiv decriptarea unei baze de date cu 100 de înregistrări în aproximativ 0,3 secunde.

Contribuția 4: Prelucrarea securizată a datelor

Articole:

- *Privacy in machine learning algorithms* [15]
- *Short text classification* [16]

Am creat un algoritm de clasificare pentru texte scurte, inclusiv seturi de date, care conțin câteva cuvinte tehnice proeminente și care sunt greu de clasificat de algoritmii de învățare automată nesupravegheați de ultimă generație. Am comparat soluția noastră cu căutarea manuală și am găsit câteva rezultate pozitive. Avem câteva potriviri care nu conțin aceleași cuvinte cheie ca cele căutate, ci sinonime și am găsit rezultate relevante care au fost ignorate de motorul de căutare manuală. Am împărțit întreaga bază de date în cinci categorii și scopul nostru a fost să conectăm fiecare intrare la una dintre categorii. Am aflat că intrările legate de „energia solară" sunt ușor de clasificat, deoarece majoritatea intrărilor sunt legate de observațiile solare. Cel mai dificil subiect de examinat este potențialul solar. Conține termeni mai generali, cum ar fi „potențial", care duce la mai multe rezultate fals pozitive.

După ce am demonstrat că putem obține informații din orice tip de date, ne-am concentrat pe adăugarea de confidențialitate peste algoritmii de învățare automată. Am pornit de la o tehnologie existentă, Federated Learning, care asigură confidențialitatea și am desfășurat mai multe experimente de clasificare a textului pentru a vedea acuratețea și penalizările de performanță în comparație cu o abordare centralizată. Am derulat testele cu doi executori privați pentru două subiecte complet diferite și pentru două subiecte similare și am luat în considerare texte de lungimi diferite, variind de la 5000 la 1000 de cuvinte. Am găsit rezultate similare atât pentru abordarea privată, cât și pentru cea centralizată și am remarcat o penalizare mai mare în cazul textelor mai scurte. Apoi, am rulat algoritmul pe un set de date de spam SMS, unde am constatat o creștere a penalizării pentru învățarea federată. Am examinat rezultatele și am constatat că penalizarea principală este cauzată de trimiterea de informații între lucrători și serverul central. Am ajuns la concluzia că acest aspect este diferit în lumea reală. Cu miliarde de dispozitive conectate, care pot servi ca lucrători, această limitare poate fi atenuată.

8 BIBLIOGRAPHY

[1] [Online]. Available: https://www.wired.com/story/the-man-who-saw-the-dangers-of-cambridge-analytica/. [Accessed November 201].

[2] N. Gruschka, V. Mavroeidis, K. Vishi and M. Jensen, "Privacy issues and data protection in big data: a case study analysis under GDPR," in *IEEE International Conference on Big Data, pp.5027-5033*, 2018.

[3] [Online]. Available: https://www.embroker.com/blog/cyber-attack-statistics/. [Accessed November 2021].

[4] [Online]. Available: https://www.securitymagazine.com/articles/95723-data-privacy-in-the-era-of-covid-19-vaccine-rollouts. [Accessed November 2021].

[5] C. Dwork, "Differential privacy: A survey of results," in *International conference on theory and applications of models of computation*, Springer, Berlin, Heidelberg, 2008.

[6] [Online]. Available: https://analyticsindiamag.com/google-comes-up-with-a-new-differentially-private-clustering-algorithm/. [Accessed November 2021].

[7] [Online]. Available: https://www.apple.com/privacy/docs/Differential_Privacy_Overview.pdf. [Accessed November 2021].

[8] [Online]. Available: https://payu.in/blog/the-big-6-steps-of-big-data-explained/. [Accessed November 2021].

[9] I. M. Florea, R. Rughinis, L. Ruse and D. Dragomir, "Survey of standardized protocols for the Internet of Things," in *21st International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, 2017.

[10] I. M. Florea, L. C. Ruse and R. Rughinis, "Challenges in security in Internet of Things," in *16th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, 2017.

[11] I. M. Florea, D. Vornicu, J. A. Vaduva and R. Rughinis, "Teaching privacy through the development and testing of a location obfuscation solution," in *The International Scientific Conference eLearning and Software for Education*, Bucharest, 2020.

[12] I. M. Florea, D. Vornicu, S. D. Ciocirlan and R. Rughinis, "Location privacy for non-stationary users," *University Politehnica of Bucharest Scientific Bulletin,* vol. 83, no. 4, 2021.

[13] I. M. Florea, S. D. Ciocirlan and I. Dura, "Practical analysis of searchable encryption strategies for financial architecture," in *9th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, 202.

[14] I. M. Florea, G. Ghinita and R. Rughinis, "Sharing of Network Flow Data across Organizations using Searchable Encryption," in *23rd International Conference on Control Systems and Computer Science (CSCS)*, Bucharest, 2021.

[15] I. M. Florea, M. Constantin and S. D. Ciocirlan, "Privacy in machine learning algorithms," in *20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, 2021.

[16] I. M. Florea, R. Rughinis and S. D. Ciocirlan, "Short text classification," in *20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, Bucharest, 2021.

[17] [Online]. Available: https://hadoop.apache.org/. [Accessed February 2022].

[18] [Online]. Available: https://auth0.com/blog/the-7-most-common-types-of-cybersecurity-attacks-in-2021/. [Accessed November 2021].

[19] [Online]. Available: https://www.the-next-tech.com/blockchain-technology/how-much-data-is-produced-every-day-2019/. [Accessed November 2021].

[20] [Online]. Available: https://www.computerweekly.com/news/252492564/Belgian-security-researcher-hacks-Tesla-with-Raspberry-Pi. [Accessed November 2021].

[21] J. Granjal, E. Monteiro and J. Sa SIlva, "Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues," *IEEE Communications Surveys & Tutorials,* vol. 17, no. 3, pp. 1294-1312, 2015.

[22] K. N. Montenegro G., J. Hui and C. D., "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, 2017.

[23] S. M. Sajjad and M. Yousaf, "Security analysis of IEEE 802.15. 4 MAC in the context of Internet of Things (IoT)," in *Conference on Information Assurance and Cyber Security (CIACS)*, 2014.

[24] P. M., R. J., M. P. and L. S., "Performance study of IEEE 802.15.4 using measurements and simulations," in *IEEE Wireless Communications and Networking Conference*, 2006.

[25] K. N., G. Montenegro and C. Schumacher, "IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals," in *RFC 4919*, 2007.

[26] P. THUBERT and J. HUI, "RFC 6282, Compression Format for IPv6 Datagrams over IEEE 802.15. 4-Based Networks," Internet Engineering Task Force, Fremont, CA, USA.

[27] ALLIANCE, LoRa, " LoRaWAN What is it?-A Technical Overview of LoRa and LoRaWAN," 2015.

[28] A. J. Jara, M.-J. Pedro and A. Skarmeta, "Light-weight multicast DNS and DNS-SD (lmDNS-SD): IPv6-based resource and service discovery for the web of things," in *Sixth international conference on innovative mobile and internet services in ubiquitous computing*, 2012.

[29] S. Cheshire and M. Krochmal, "Multicast DNS," RFC 6762, 2014.

[30] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhar and M. Ayyash, "Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications," *IEEE communications surveys & tutorials,* vol. 17, no. 4, pp. 2347-2376, 2014.

[31] S. Cheshire and M. Krochmal, "DNS-based service discovery," RFC 6763, 2013.

[32] R. Klauck and M. Kirsche, "Bonjour contiki: A case study of a DNS-based discovery service for the internet of things," in *International Conference on Ad-Hoc Networks and Wireless*, Springer, Berlin, Heidelberg, 2012.

[33] I. Ishaq, D. Carels, G. Teklemariam, J. Hoebeke, F. Van den Abeele, E. De Poorter, I. Moerman and P. Demeester, "IETF Standardization in the Field of the Internet of Things (IoT): A Survey," *Jurnal of Sensor and Actuator Networks,* vol. 2, no. 2, pp. 235-287, 2013.

[34] Z. Shelby, K. Hartke and C. Bormann, "The constrained application protocol (CoAP)," 2014.

[35] C. Bormann, A. P. Castellani and Z. Shelby, "Coap: An application protocol for billions of tiny internet nodes," *IEEE Internet Computing,* vol. 16, no. 2, pp. 62-67, 2012.

[36] E. Rescorla and M. Nagendra, "Datagram transport layer security," 2016.

[37] D. McGrew and D. Bailey, " Aes-ccm cipher suites for transport layer security (tls)," *RFC 6655,* pp. 1-6, 2016.

[38] "SECG-Elliptic Curve Cryptography-SEC 1," [Online]. Available: http://www.secg.org. [Accessed 17 February 2017].

[39] S. L. Keoh, O. Garcia Morchon, K. S. Sandeep and E. Dijk, "DTLS-based Multicast Security for Low-Power and Lossy Networks (LLNs)," draft-keoh-tls-multicast-security-00, 2014.

[40] S. L. Keoh, S. K. Sandeep and H. Tschofenig, "Securing the internet of things: A standardization perspective," *IEEE Internet of things Journal,* vol. 1, no. 3, pp. 265-275, 2014.

[41] K. Hartke, "Practical Issues with Datagram Transport Layer Security in Constrained Environments," draft-hartke-dice-practical-issues-00, 2013.

[42] S. Santesson and T. Hannes, "Transport layer security (TLS) cached information extension," draft-ietf-tls-cached-info-23.txt, 2014.

[43] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)," draft-ietf-tls-oob-pubkey-11, 2014.

[44] S. Keoh, S. Kumar and Z. Shelby, "Profiling of DTLS for CoAP-Based IoT Applications," draft-keoh-dice-dtls-profile-iot-00, 2013.

[45] D. Locke, "Mq telemetry transport (mqtt) v3. 1 protocol specification," *IBM developerWorks Technical Library,* vol. 15, 2010.

[46] U. Hunkeler, H. L. Truong and A. Stanford-Clark, "MQTT-S — A publish/subscribe protocol for Wireless Sensor Networks," in *3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE'08)*, 2008.

[47] A. Stanford-Clark and H. L. Truong, "Mqtt for sensor networks (mqtt-sn) protocol specification," *International business machines (IBM) Corporation version,* vol. 1, no. 2, pp. 1-28, 2013.

[48] D. Thangavel, X. Ma, A. Valera, H.-X. Tan and C. Keng-Yan Tan, "Performance evaluation of MQTT and CoAP via a common middleware," in *IEEE ninth international conference on intelligent sensors, sensor networks and information processing (ISSNIP)* , 2014.

[49] [Online]. Available: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/426248/Acquisition_and_Disclosure_of_Communications_Data_Code_of_Practice_March_2015.pdf. [Accessed February 2022].

[50] M. Andres, N. Bordenabe, C. Konstantinos and C. Palamidessi, "Geo-Indistinguishability: Differential Privacy for Location-Based Systems," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, 2013.

[51] N. Bordenabe, C. Konstantinos and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, 2014.

[52] L. Sweeney, "k-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* vol. 10, no. 5, pp. 557-570, 2002.

[53] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg and D. Boneh, "Location Privacy via Private Proximity Testing," *NDSS,* vol. 11, 2011.

[54] R. Shokri, C. Troncoso, C. Diaz, J. Freudiger and J.-P. Hubaux, "Unraveling an Old Cloak: k-anonymity for Location Privacy," in *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, 2010.

[55] A.-M. Olteanu, K. Huguenin, R. Shokri, M. Humbert and J.-P. Hubaux, "Quantifying Interdependent Privacy Risks with Location," *IEEE Transactions on Mobile Computing,* vol. 16.3, pp. 829-842, 2017.

[56] [Online]. Available: https://www.wired.com/2007/12/why-anonymous-data-sometimes-isnt/. [Accessed November 2021].

[57] O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," *Journal of the ACM,* vol. 43, no. 3, pp. 431-473, 1996.

[58] C. Bosch, P. Hartel, W. Jonker and A. Peter, "A Survey of Provably Secure Searchable Encryption," *ACM Computing Surveys,* vol. 47, no. 2, pp. 1-51, 2015.

[59] D. Boneh and B. Waters, "Conjunctive, Subset, and Range Queries on Encrypted Data," in *4th Theory of Cryptography Conference*, Amsterdam, 2007.

[60] D. Boneh, E.-J. Goh and K. Nissim, "Evaluating 2-DNF Formulas on Ciphertexts," in *Proceedings of Theory of Cryptography Conference*, 2005.

[61] V. Iovino and G. Persiano, "Hidden-vector encryption with groups of prime order," in *International Conference on Pairing-Based Cryptography*, Berlin, 2008.

[62] J. Katz, A. Sahai and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Annual international conference on the theory and applications of cryptographic techniques*, Berlin, 2008.

[63] S. Kim, L. Kewi, A. Mandal, H. Montgomery, A. Roy and D. J. Wu, "Function-Hiding Inner Product Encryption Is Practical," in *International Conference on Security and Cryptography for Networks*, 2018.

[64] R. Bost, B. Minaud and O. Ohrimenko, "Forward and Backward Private Searchable Encryption from Constrained Cryptographic Primitives," in *CCS '17: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

[65] C. David, S. Jarecki, J. Charanjit, K. Hugo, R. Marcel-Cătălin and M. Steiner, "Highly-Scalable Searchable Symmetric Encryption with Support for Boolean Queries," in *CRYPTO 2013: Advances in Cryptology*, Berlin, 2013.

[66] S. Lai, S. Patranabis, A. Sakzad, J. K. Liu, D. Mukhopadhyay, R. Steinfeld, S.-F. Sun, D. Liu and C. Zuo, "Result Pattern Hiding Searchable Encryption for Conjunctive Queries," in *CCS '18: Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018.

[67] S. Lai, S. Patranabis, A. Sakzad, J. K. Liu, D. Mukhopadhyay, R. Steinfeld, S.-F. Sun, D. Liu and C. Zuo, "Result Pattern Hiding Searchable Encryption for Conjunctive Queries," in *ACM SIGSAC Conference on Computer and Communications Security (CCS '18)*, Toronto, On, Canada, 2018.

[68] B. H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM,* vol. 13, no. 7, pp. 422-426, 1970.

[69] F. Sky, S. Jarecki, H. Krawczyk, N. Quan, R. Marcel and M. Steiner, "Rich Queries on Encrypted Data: Beyond Exact Matches," in *European symposium on research in computer security*, 2015.

[70] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, 2016.

[71] "OpenMined Community. Pysyft Library.," [Online]. Available: https://github.com/OpenMined/PySyft. [Accessed 09 May 2021].

[72] "OpenMined Community. Pygrid," [Online]. Available: https://github.com/OpenMined/PyGrid. [Accessed 21 June 2021].

[73] "OpenMined Community. Kotlinsyft.," [Online]. Available: https://github.com/OpenMined/KotlinSyft. [Accessed 21 June 2021].

[74] "OpenMined Community. Swiftsyft," [Online]. Available: https://github.com/OpenMined/SwiftSyft. [Accessed 21 June 2021].

[75] "OpenMined Community. syft.js," [Online]. Available: https://github.com/OpenMined/syft.js. [Accessed 21 June 2021].

[76] D. Heimbigner and D. Mcleod, " A federated architecture for information management," in *ACM Transactions on Information Systems (TOIS)*, 1985.

[77] A. MACHANAVAJJHALA, X. HE and M. HAY, " Differential privacy in the wild: A tutorial on current practices & open challenges," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017.

[78] T. Yang, G. Andrew, H. Eichner, H. Sun, W. Li, N. Kong, D. Ramage and F. Beaufays, "Applied Federated Learning: Improving Google Keyboard Query Suggestions," in *arXiv preprint arXiv:1812.02903*, 2018.

[79] T. Ryffel, A. Trask, M. Dahl, B. Wagner, J. Mancuso, D. Rueckert and J. Passerat-Palmbach, "A generic framework for privacy preserving deep learning," *arXiv preprint arXiv:1811.04017,* 2018.

[80] C. Dwork and R. Aaron, "The algorithmic foundations of differential privacy," *Foundations and Trends in Theoretical Computer Science,* vol. 9, pp. 211-407, 2014.

[81] T. K. Landauer, P. W. Foltz and L. Darrell, " An introduction to latent semantic analysis," *Discourse processes,* vol. 25, no. 2-3, pp. 259-284, 1998.

[82] X. Yan, J. Guo, Y. Lan and X. Cheng, "A biterm topic model for short texts," in *Proceedings of the 22nd international conference on World Wide Web*, 2013.

[83] T. MINKA, "Estimating a Dirichlet distribution," 2000.

[84] K. Sharma and T. Suryakanthi, "Smart System: IoT for University," in *International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2015.

[85] M. Nati, G. Alexander, A. Hamidreza and H. William, "SmartCampus: A user-centric testbed for Internet of Things experimentation," in *16th International symposium on wireless personal multimedia communications (WPMC)*, 2013.

[86] "EU FP7 ICT WISEBED project," [Online]. Available: http://www.wisebed.eu/. [Accessed 12 February 2017].

[87] [Online]. Available: https://www.microsoft.com/en-us/download/details.aspx?id=52367. [Accessed May 2021].

[88] P. Wightman, W. Coronell, D. Jabba, M. Jimeno and M. Labrador, "Evaluation of Location Obfuscation techniques for privacy in location based information systems," in *IEEE Third Latin-American Conference on Communications*, 2011.

[89] L. Yu, L. Ling and C. Pu, "Dynamic Differential Location Privacy with Personalized Error Bounds," in *NDSS*, 2017.

[90] K. Fawaz and S. K. G., "Location privacy protection for smartphone users," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014.

[91] Y. Lu, "Privacy-preserving Logarithmic-time Search on Encrypted Data," in *NDSS*, 2012.

[92] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Annual international cryptology conference*, Berlin, 2001.

[93] G. Ghinita and R. Rughinis, "An efficient privacy-preserving system for monitoring mobile users: making searchable encryption practical," in *Proceedings of the 4th ACM conference on Data and application security and privacy*, 2014.

[94] "Stanford Crypto," [Online]. Available: https://crypto.stanford.edu/pbc/thesis.html. [Accessed 17 June 2020].

[95] "GMP Libraries for C," [Online]. Available: https://gmplib.org. [Accessed 17 June 2020].

[96] [Online]. Available: https://github.com/MonashCybersecurityLab/HXT. [Accessed 17 June 2020].

[97] [Online]. Available: https://www.cs.cmu.edu/afs/cs/academic/class/15492-f07/www/pthreads.html. [Accessed 20 February 2022].

[98] [Online]. Available: https://www.kaggle.com/uciml/sms-spam-collection-dataset. [Accessed 20 February 2022].

[99] S. Caldas, J. Konečny, H. B. McMahan and T. Ameet, "Expanding the reach of federated learning by reducing client resource requirements," in *arXiv preprint arXiv:1812.07210*, 2018.

[100] S. Vijayarani, M. J. Ilamathi and M. Nithya, "Preprocessing techniques for text mining-an overview," in *International Journal of Computer Science & Communication Networks*, 2015.

[101] A. G. Jivani, "A comparative study of stemming algorithms," in *Int. J. Comp. Tech. Appl*, 2011.

[102] S. Deepika, "Stemming Algorithms, A Comparative Study and their Analysis," in *International Journal of Applied Information Systems (IJAIS) –ISSN*, New York, 2012.

[103] D. Harman, "How effective is suffixing?," *Journal of the American Society for Information Science,* vol. 42, pp. 7-15, 1991.

[104] M. F. Porter, "An algorithm for suffix stripping," *Program,* 1980.

[105] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and D. Jeff, Advances in neural information processing systems, 2013.

[106] K. W. Church, "Word2Vec," *Natural Language Engineering,* vol. 23, no. 1, pp. 155-162, 2016.

[107] "Zenodo," [Online]. Available: https://zenodo.org/. [Accessed May 2020].

[108] "Google Scholar," [Online]. Available: https://scholar.google.com/. [Accessed May 2020].

[109] [Online]. Available: https://radimrehurek.com/gensim/models/word2vec.html. [Accessed May 2020].

[110] T. K. LANDAUER, P. W. FOLTZ and D. LAHAM, "An introduction to latent semantic analysis," *Discourse Processes,* vol. 25, no. 2-3, pp. 259-284, 1998.

[111] [Online]. Available: http://www.bom.gov.au/jsp/ncc/climate_averages/solar-exposure/index.jsp?period=an#maps. [Accessed May 2020].

[112] [Online]. Available: https://www.giz.de/en/worldwide/17995.html. [Accessed February 2022].

[113] H. Baali, H. Djelouat, A. Amira and F. Bensaali, "Empowering Technology Enabled Care Using IoT and Smart Devices: A Review," *IEEE Sensors Journal,* vol. 18, no. 5, pp. 1790-1809, 2018.

[114] A. Haroon, S. Akram, M. A. Shah and A. Wahid, "E-Lithe: A Lightweight Secure DTLS for IoT," in *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, 2017.

[115] A. K. Jain and R. C. Dubes., Algorithms for Clustering Data, Upper Saddle River: Prentice-Hall, Inc., 1988.

[116] "Kernel panic! What are Meltdown and Spectre, the bugs affecting nearly every computer and device?," techcrunch.com, 2018. [Online]. Available: https://techcrunch.com/2018/01/03/kernel-panic-what-are-meltdown-and-spectre-the-bugs-affecting-nearly-every-computer-and-device. [Accessed 14 02 2018].

[117] E. Rogers, "Understanding Buck-Boost Power Stages in Switch Mode Power Supplies," Texas Instruments, 2007.

[118] J. Silva-Martinez, "ELEN-325. Introduction to Electronic Circuits: A Design Approach," 2008. [Online]. Available: http://www.ece.tamu.edu/~spalermo/ecen325/Section%20III.pdf.

[119] K. Bonawitz, H. Eichner, W. Grieskamp, D. Huba, A. Ingerman, V. Ivanov, C. Kiddon, J. Konečnỳ, S. Mazzocchi and M. H. Brendan, "Towards federated learning at scale: System design," in *arXiv preprint arXiv:1902.01046*, 2019.