

Universitatea “Alexandru Ioan Cuza” din Iași

Facultatea de Informatică



LUCRARE DE LICENȚĂ

propusă de

Paul-Vlăduț Sima

Sesiunea: iunie-iulie,2021

Coordonator științific

Asist.Dr. Anca-Maria Nica

UNIVERSITATEA “ALEXANDRU IOAN CUZA” DIN IAȘI

FACULTATEA DE INFORMATICĂ

Utilizarea CP-ABE pentru aplicații practice

Paul-Vlăduț Sima

Sesiunea: iunie-iulie, 2021

Coordonator științific

Asist.Dr. Anca-Maria Nica

Cuprins

Contribuții	8
CAPITOLUL 1 DETALII TEHNICE	9
1.1 Descrierea schemei CP-ABE	9
1.2 Definiții	10
1.3 Modelul de securitate	12
CAPITOLUL 2 IMPLEMENTAREA APLICAȚIEI	20
2.1 Modul de funcționare al aplicației.....	20
2.2 Tehnologii folosite.....	25
CAPITOLUL 3 CONCLUZII ȘI DIRECȚII VIITOARE	27
Bibliografie.....	28

Introducere

Există o tendință ca datele sensibile ale utilizatorilor să fie stocate de terți pe Internet. De exemplu, e-mailul personal, datele și preferințele personale sunt stocate pe site-uri de portal web precum Google și Yahoo. Având în vedere varietatea, cantitatea și importanța informațiilor stocate pe aceste site-uri, există motive de îngrijorare cu privire la faptul că datele personale vor fi compromise. Această îngrijorare este escaladată de creșterea atacurilor recente și de presiunea juridică cu care se confruntă astfel de servicii. O metodă pentru a atenua unele dintre aceste probleme este stocarea datelor în formă criptată. Astfel, dacă stocarea este compromisă, cantitatea de pierderi de informații va fi limitată. Un dezavantaj al criptării datelor este că limitează sever capacitatea utilizatorilor de a partaja în mod selectiv datele criptate la un nivel foarte detaliat. Să presupunem că un anumit utilizator dorește să acorde acces de decriptare unei părți la toate jurnalele sale de trafic pe internet pentru toate intrările dintr-un anumit interval de date care au avut o adresă IP dintr-o anumită subrețea. Utilizatorul fie trebuie să acționeze ca intermediar și să decripteze toate intrările relevante, fie trebuie să dea părții cheia sa de decriptare privată și, astfel, să-i permită să aibă acces la toate intrările. Nici una dintre aceste opțiuni nu este deosebit de atrăgătoare.

Dezvoltările recente în rețelele de senzori fără fir, Internet of Things (IoT) și cloud computing ridică probleme tot mai mari în ceea ce privește controlul accesului. Tehnicile standard de control al accesului, precum controlul accesului discreționar, controlul accesului obligatoriu sau controlul accesului bazat pe roluri, se dovedesc a fi inadecvate în astfel de cazuri. De exemplu, controlul accesului discreționar nu este potrivit pentru rețele de mari dimensiuni cu cerințe ridicate de securitate, în principal deoarece nu oferă niciun mecanism sau metodă de gestionare a controlului accesului necorespunzător: dacă software-ul nu reușește să restricționeze utilizatorul de la permisiunile predefinite, atunci orice hacker poate pătrunde în sistem, poate avea acces la fișierele confidențiale și poate efectua, de asemenea, toate acțiunile cum ar fi citirea, scrierea sau ștergerea. Nici controlul accesului nu se descurcă mai bine în astfel de cazuri. De exemplu, este dificil de implementat în sistemele cloud, deoarece nu acceptă separarea de atribuții, delegare sau moștenire. Deși controlul accesului bazat pe roluri ameliorează unele dintre problemele de securitate cu celelalte două, acesta nu este încă foarte potrivit pentru cloud computing, deoarece nu se

adaptează ușor la sistemele cu un număr mare de utilizatori și roluri, în care rolurile utilizatorului se schimbă frecvent. Mai mult, este dificil să extindeți controlul accesului bazat pe roluri pe domenii administrative, deoarece este dificil să decideți privilegiile unui rol.

(Sahai A., 2005) a făcut câțiva pași inițiali pentru rezolvarea acestei probleme prin introducerea conceptului de criptare bazată pe atribute (ABE). Într-un sistem ABE, cheile și textele cifrate ale unui utilizator sunt etichetate cu seturi de atribute descriptive și o anumită cheie poate decripta un anumit text cifrat numai dacă există o potrivire între atributele textului cifrat și cheia utilizatorului. Criptosistemul Sahai și Waters a permis decriptarea atunci când cel puțin k atribute s-au suprapus între un text cifrat și o cheie privată. În timp ce această primitivă s-a dovedit a fi utilă pentru criptarea tolerantă la erori cu biometrie, lipsa de expresibilitate pare să îi limiteze aplicabilitatea la sisteme mai mari.

Într-un sistem de criptare bazat pe atribute, textele cifrate nu sunt neapărat criptate pentru un anumit utilizator, ca în criptografia cu cheie publică tradițională. În schimb, atât cheile private, cât și textele cifrate ale utilizatorilor vor fi asociate cu un set de atribute sau cu o politică asupra atributelor. Un utilizator este capabil să decripteze un text cifrat dacă există o „potrivire” între cheia sa privată și textul cifrat. În sistemul lor original, Sahai și Waters au prezentat un sistem Threshold ABE în care textele cifrate erau etichetate cu un set de atribute S și cheia privată a unui utilizator era asociată atât cu un parametru k , cât și cu un alt set de atribute S' . Pentru ca un utilizator să decripteze un text cifrat, cel puțin k atribute trebuie să se suprapună între textul cifrat și cheile sale private. Una dintre principalele motivații inițiale pentru aceasta a fost de a proiecta o schemă de criptare bazată pe identitate tolerantă la erori (sau Fuzzy) [(Shamir, 1979), (D. Boneh, 2003)] care să poată utiliza identități biometrice.

Principalul dezavantaj al limitelor sistemului ABE (Sahai A., 2005) este că semantica pragului nu este foarte expresivă și, prin urmare, este limitativă pentru proiectarea sistemelor mai generale. Goyal și colab. a introdus ideea unui sistem de criptare bazat pe atributul politicii cheie mai general. În construcția lor, un text cifrat este asociat cu un set de atribute, iar cheia unui utilizator poate fi asociată cu orice arbore ce reprezintă o structură de acces monotonă. Construcția lui Goyal și colab. poate fi privită ca o extensie a tehnicilor Sahai-Waters unde în loc să încorporeze o schemă de partajare secretă (Shamir, 1979) în cheia privată, autoritatea încorporează o schemă de împărțire a secretelor mai generală pentru o structură de acces

monotonă. Goyal și colab. a sugerat, de asemenea, posibilitatea unei scheme ABE de text cifrat (CP-ABE) , dar nu a oferit construcții.

(Matthew Pirretti, 2006) a oferit o implementare a sistemului de criptare ABE threshold, a demonstrat diferite aplicații ale schemelor de criptare bazate pe atribute și a abordat mai multe noțiuni practice, cum ar fi revocarea cheii. În lucrările recente, (Chase, 2007) a oferit o construcție pentru un sistem de criptare bazat pe atribute cu mai multe autorități, în care fiecare autoritate ar administra un domeniu diferit de atribute. Principala provocare în crearea ABE cu mai multe autorități este de a preveni atacurile de coluziune între utilizatori care obțin componente cheie de la diferite autorități. În timp ce sistemul Chase a folosit sistemul ABE threshold ca sistem ABE subiacent la fiecare autoritate, problema ABE cu mai multe autorități este în general ortogonală pentru a găsi sisteme ABE mai expresive.

Proprietatea definitorie a sistemelor de criptare bazată pe atribute este rezistența lor la atacurile de coluziune. Această proprietate este esențială pentru construirea sistemelor de control al accesului criptografic; în caz contrar, este imposibil să se garanteze că un sistem va prezenta proprietățile de securitate dorite, deoarece vor exista atacuri devastatoare ale unui atacator care reușește să dețină câteva chei private. Deși am putea lua în considerare sistemele ABE cu diferite nuanțe de expresibilitate, lucrările anterioare [(Sahai A., 2005), (Vipul Goyal, 2006)] au arătat clar că rezistența la coluziune este o proprietate necesară a oricărui sistem ABE.

Înainte de introducerea criptării bazate pe atribute, au existat alte sisteme care au încercat să abordeze controlul accesului la datele criptate [(Smart, 2003), (Robert W. Bradshaw, 2004)] utilizând scheme de partajare a secretelor [(Mitsuru Ito, 1989), (Ernest F. Brickell, 1991), (Shamir, 1979), (Blakley), (Benaloh J., 1990)] combinat cu criptare bazată pe identitate; cu toate acestea, aceste scheme nu au abordat rezistența la atacurile de coluziune. Recent, Kapadia, Tsang și Smith (Apu Kapadia, 2007) au oferit o schemă de control al accesului criptografic care folosea servere proxy. Munca lor a explorat noi metode pentru utilizarea serverelor proxy pentru a ascunde politicile și a utiliza controlul accesului non-monotonic pentru universuri mici de atribute. Menționăm că, deși ei au numit acest sistem o formă de CP-ABE, schema nu are proprietatea de rezistență la coluziune. Ca atare, credem că munca lor nu ar trebui luată în considerare în clasa sistemelor de criptare bazate pe atribute din cauza lipsei de securitate împotriva atacurilor de coluziune.

În această lucrare voi demonstra utilizarea schemei CP-ABE în implementarea aplicațiilor practice. Acest tip de criptare al datelor poate fi folosit în orice tip de aplicație care necesită

granularizarea controlului accesului pentru un număr mare de utilizatori. Prin adaptarea mulțimii de attribute pentru contextul în care se folosește, structura bazei de date și schema ABE utilizată, criptarea bazată pe attribute poate personaliza controlul accesului. Astfel, doar utilizatorii autorizați (care dețin attributele cu care a fost criptată informația) pot decripta datele. În această lucrare aplicația implementată de mine este folosită pentru gestionarea unui Dosar Electronic de Sănătate (în engleză Personal Health Record), însă nu este limitat la asta. Se pot gestiona mai multe tipuri de aplicații prin implementarea unui nou client și conectarea la altă bază de date.

Dosarul Electronic de Sănătate (DES) este un fel de aplicație bazată pe cloud care are o capacitate puternică de a stoca, partaja și analiza stările de sănătate ale utilizatorilor, istoricul bolilor, medicamente etc. oferind astfel funcții diversificate. În prezent, sistemele PHR sunt utilizate pe scară largă în reabilitarea bolilor, prevenirea bolilor și tratamentul medical. Datorită capacității sale uriașe de date, este convenabil pentru utilizatori să obțină ceea ce doresc. Dar sistemele conțin în mod natural informații sensibile ale pacienților, oricine poate accesa aceste date dacă nu există supraveghere. De exemplu, furnizorii de servicii care nu prezintă credibilitate și utilizatorii neautorizați nu ar trebui să poată accesa aceste date. În consecință, unele tehnici orientate spre securitate sunt esențiale a fi aplicate pentru a supraveghea accesul necorespunzător.

Schema CP-ABE implementată este descrisă de John Bethencourt John Bethencourt, Amit Sahai și Brent Waters în (John Bethencourt, 2007) și în (Waters, 2008).

Contribuții

Contribuția mea personală este îmbunătățirea granularității controlului accesului oferită de CP-ABE prin derivarea cheii generate pentru utilizator și a regulii logice folosită la criptarea informației.

Pentru a înțelege mai bine această modificare este necesar să știm structura CP-ABE. Această schemă de criptare este compusă din patru algoritmi: **Setup**, **Encrypt**, **KeyGen** și **Decrypt**.

- **Setup(λ , U).** Acest algoritm are ca input parametrul de securitate și descrierea mulțimii de attribute. Are ca output parametrii publici PK și o cheie secretă MK (Master Key).
- **Encrypt(PK, M, A).** Algoritmul de criptare primește ca input parametrii publici PK, un mesaj M și o structură de acces A peste mulțimea de attribute U. Are ca output un text criptat CT astfel încât doar utilizatorii care dețin attributele care satisfac structura de acces A. Contribuția mea la acest pas este de a adăuga un atribut adițional, ce reprezintă un identificator unic al deținătorului informației (acest atribut este același cu atributul folosit pentru derivarea cheii utilizatorilor).
- **Key Generation(MK, S).** Algoritmul de generare a cheilor are ca input cheia secretă MK și un set de attribute S care descriu cheia. Are ca output o cheie privată pentru un utilizator SK. Contribuția mea la acest pas este de a deriva cheia adăugând un atribut ce reprezintă un identificator unic al utilizatorului.
- **Decrypt(PK, CT, SK).** Algoritmul de decriptare are ca input PK, textul criptat CT, care conține o structură de acces A și o cheie privată pentru setul de attribute S, SK. Dacă setul de attribute S satisface S, atunci CT este decriptat și algoritmul returnează textul decriptat M.

Această modificare permite delegarea unui singur utilizator de a decripta un mesaj ce a fost criptat folosind atributul unic al acestuia. De asemenea, deținătorul datelor poate decripta mesajul chiar dacă cheia lui privată nu satisface structura de acces A.

Capitolul 1 Detalii tehnice

1.1 Descrierea schemei CP-ABE

Lipsa satisfacției cu demonstrarea modelului de grup generic a motivat problema găsirii unui sistem CP-ABE expresiv sub un model mai solid. Au existat abordări multiple în această direcție.

În primul rând, putem vedea construcția (Sahai A., 2005) cel mai „natural” ca Key-Policy ABE pentru o poartă de prag. În lucrarea lor, Sahai și Waters descriu cum să realizeze Ciphertext-Policy ABE pentru porțile de prag prin „altoirea” așa-numitelor „attribute fictive” peste sistemul lor de bază. În esență, au transformat un sistem KP-ABE într-unul CP-ABE cu expresivitatea unei singure porți de prag. (Ling Cheung, 2007) oferă o construcție directă pentru construirea unei politici cu o singură poartă AND și sub presupunerea bilineară Diffie-Hellman. Abordarea lor are dezavantajele că permite doar un număr fix de attribute de sistem și este limitată la o poartă AND (nu permite praguri). Retrospectiv, aceste două limitări îl fac de fapt mai puțin expresiv decât transformarea SW, deși acest lucru nu a fost neapărat imediat evident.

(Goyal V., 2008) au generalizat abordarea transformațională pentru a arăta cum să transforme un sistem KP-ABE într-unul CP-ABE folosind ceea ce ei numesc un „arbore de acces universal”. În special, au furnizat o mapare pe un arbore de acces „universal” (sau complet) cu formule de până la adâncime "d" constând din porți de prag cu dimensiunea de intrare "m", unde "m" și "d" sunt alese de algoritmul de configurare. Au aplicat o abordare similară a „atributului fals”.

Pentru a suporta o formulă generală de acces de dimensiunea n, schema lor se traduce mai întâi într-o formulă echilibrată. În cadrul tehnicilor standard, o formulă de dimensiunea n poate fi „echilibrată” astfel încât orice formulă (arbore) de dimensiunea n poate fi acoperită de un arbore complet de dimensiune aproximativ $O(n^{3.42})$. Munca lor a fost primul rezultat de fezabilitate pentru CP-ABE expresiv sub o ipoteză non-interactivă. Din păcate, parametrii textului cifrat și dimensiunile cheilor private cauzează complexitatea de criptare și decriptare să explodeze (în cel mai rău caz) cu un factor $n^{3.42}$ care îi limitează utilitatea în practică. De exemplu, într-un sistem cu attribute U definite și n noduri, cifra generală a textului cifrat va fi

aproximativ un factor de $U * n^{2.42}$ mai mare decât cel al sistemului BSW. Pentru a da un exemplu concret, pentru parametrii modești ai dimensiunii universului $U = 100$ de attribute și o formulă de 20 de noduri, factorul de explozie relativ la BSW este de aproximativ 140000.

Prezentăm o nouă metodologie pentru realizarea sistemelor ABE Ciphertext-Policy dintr-un set general de structuri de acces în modelul standard sub ipoteze concrete și neinteractive. Atât costurile textului cifrat, cât timpul de criptare sunt proportionale cu $O(n)$ unde n este dimensiunea formulei. În plus, timpul de decriptare este proporțional cu numărul de noduri.

1.2 Definiții

- Definiția 1 (Structura de acces) Fie $\{ P_1, P_2, P_3, \dots, P_n \}$ un set de părți. O mulțime $A \subseteq 2 \{P_1, P_2, \dots, P_n\}$ este monotonă dacă $\forall B, C : \text{if } B \in A \text{ and } B \subseteq C \text{ then } C \in A$. O structură de acces(respectiv o structură monotonă de acces) este o mulțime(respectiv mulțime monotonă) A formată din submulțimi nevide din $\{ P_1, P_2, P_3, \dots, P_n \}$, $A \subseteq 2\{P_1, P_2, \dots, P_n\} \setminus \{\emptyset\}$. Submulțimile din A sunt numite submulțimi autorizate, și submulțimile care nu sunt în A se numesc submulțimi neautorizate.

În contextul nostru, rolul părților este luat de attribute. Astfel, structura de acces A va conține seturile autorizate de attribute. Ne restricționăm atenția la structurile de acces monotone. Cu toate acestea este, de asemenea, posibil să se realizeze (ineficient) structuri de acces generale care folosesc tehnicile noastre prin faptul negația unui atribut este un atribut separat cu totul. Astfel, numărul de attribute din sistem va fi dublat. De acum înainte, dacă nu se specifică altfel, printr-un acces structură ne referim la o structură de acces monotonă.

- Definiția 2 (Algoritmi) O schemă CP-ABE consistă în patru algoritmi fundamentali: Setup, Encrypt, KeyGen, and Decrypt. În plus, există opțiunea de a adăuga încă un algoritm, Delegate.
 1. Setup. Algoritmul de configurare nu ia nicio altă intrare decât parametrul de securitate implicit. Se afișează parametrii publici PK și o cheie master MK . Acest algoritm un grup bilinear G_0 de ordin prim p . În continuare va alege doi exponenți aleatori $\alpha, \beta \in \mathbb{Z}_p$. Cheia publică este de forma $PK = G_0, g, h = g^\beta, f = g^{1/\beta}, e(g, g)^\alpha$

și cheia master MK este $s(\beta, g^a)$. (De menționat este că f este utilizat doar pentru Delegate).

2. $\text{Encrypt}(\text{PK}, M, T)$. Algoritmul criptează un mesaj M folosind arborele T ca structură de acces. Algoritmul alege prima dată un q_x polinomial pentru fiecare nod x (inclusiv frunzele) din arborele T . Aceste polinomiale sunt alese în modul următor top-down, începând cu nodul rădăcină R . Pentru fiecare nod x din T , setează gradul d_x al polinomialului q_x să fie cu unul mai puțin decât valoarea threshold k_x al acelui nod, $d_x = k_x - 1$. Începând cu nodul rădăcină R algoritmul alege un număr aleator $s \in \mathbb{Z}_p$ și setează $q_R(0) = s$. Apoi alege încă alte d_R puncte aleatoare al polinomialei q_R pentru a completa și defini q_x . Fie Y mulțimea de noduri frunză din T . Textul criptat este construit dând ca parametru structura de acces T în felul următor: $CT = T$, $C^* = \text{Me}(g, g)^{as}$, $C = h^s$, $\forall y \in Y : Cy = g^{q_y(0)}$, $C'_y = H(\text{att}(y))^{q_y(0)}$.
3. $\text{KeyGen}(\text{MK}, S)$. Algoritmul de generare al cheilor ia ca parametru un set de attribute S și are ca output o cheie care se identifică cu acel set. Se alege un număr aleator $r \in \mathbb{Z}_p$ și apoi un număr aleator $r_j \in \mathbb{Z}_p$ pentru fiecare atribut $j \in S$. Apoi calculează o cheie în modul următor: $SK = (D = g^{(\alpha+r)/\beta}, \forall j \in S : D_j = g^{r_j} \cdot H(j)^{r_j}, D'_j = g^{r_j})$.
4. $\text{Delegate}(SK, S')$. Algoritmul de delegare primește ca parametru o cheie SK pentru un set de attribute S , și un set de attribute S' astfel încât $S' \subseteq S$. Cheia secretă este de forma $SK = (D, \forall j \in S : D_j, D'_j)$. Algoritmul alege aleator r' și $r'_k \forall k \in S'$. Apoi crează o nouă cheie secretă în modul următor: $SK' = (D' = D^{r'}, \forall k \in S' : D'_k = D_k g^{r' H(k)^{r'_k}}, D''_k = D'_k g^{r'_k})$. Cheia secretă rezultată SK' este o cheie pentru setul S' . Din moment ce algoritmul recalculează aleator cheia, ea este echivalentă cu una primită direct de la autoritate.
5. $\text{Decrypt}(CT, SK)$. Algoritmul de decriptare primește ca parametru un text criptat CT , o cheie SK și returnează textul decriptat M .

- Definiția 3 (Arborele de acces T) Fie T un arbore care reprezintă o structură de acces. Fiecare nod fără frunze al arborelui reprezintă un threshold, descris de copiii săi și o threshold. Dacă num_x este numărul de copii ai unui nod x și k_x este valoarea threshold, atunci $0 < k_x \leq \text{num}_x$. Când $k_x = 1$, poarta de prag este o poartă OR și când $k_x = \text{num}_x$,

este o poartă AND. Fiecare nod frunză x al arborelui este descris printr-un atribut și o valoare prag $k_x = 1$. Pentru a facilita lucrul cu arborii de acces, definim câteva funcții. Notăm părintele nodului x în arborele de părinte (x). Funcția $\text{att}(x)$ este definită numai dacă x este un nod frunză și denotă atributul asociat cu nodul frunzei x din copac. Arborele de acces T definește, de asemenea, o ordonare între copiii fiecărui nod, adică copiii unui nod sunt numerotați de la 1 la num . Funcția $\text{index}(x)$ returnează astfel un număr asociat nodului x . Unde valorile indexului sunt atribuite în mod unic nodurilor din structura de acces pentru o cheie dată într-un mod arbitrar.

- Definiția 4 (Cum se satisface un arbore de acces) Fie T un arbore de acces cu rădăcina r . Notăm cu T_x subarborele T înrădăcinat la nodul x . Prin urmare, T este același cu T_r . Dacă un set de atributele γ satisface arborele de acces T_x , îl denumim ca fiind $T_x(\gamma) = 1$. Calculăm $T_x(\gamma)$ recursiv după cum urmează. Dacă x este un nod fără frunze, evaluăm $T_{x'}(\gamma)$ pentru toți copiii x' de nod x . $T_x(\gamma)$ returnează 1 dacă și numai dacă cel puțin k_x copii returnează 1. Dacă x este un nod frunză, atunci $T_x(\gamma)$ returnează 1 dacă și numai dacă $\text{att}(x) \in \gamma$.

1.3 Modelul de securitate

În (Canetti R., 2004), Canetti, Halevi și Katz au oferit o construcție generică pentru criptarea cheii publice sigure CCA, folosind CPA securizat IBE și semnături puternic existențiale care nu poate fi falsificată. Ideea principală este de a asocia cheile de semnătură unice (K_v , K_s) cu fiecare operațiune de criptare. Cheia de verificare K_v este privită ca o identitate în schema IBE cu care mesajul M este criptat. Textul cifrat rezultat este apoi semnat folosind cheia de semnare K_s . Această semnătură este trimisă împreună cu textul cifrat și trebuie verificat înainte de decriptare. După cum se dovedește, se poate aplica aceeași tehnică generală la schemele ABE. În (Vipul Goyal, 2006), Goyal și colab. a oferit un CPA securizat pentru schema KP-ABE și a subliniat o extensie securizată CCA, în care mesajul M este criptat folosind un atribut suplimentar corespunzător reprezentării șirului de biți a K_v .

Această extensie se bazează pe o construcție universală mare (Sahai A., 2005) (unde se pot adăuga atribute arbitrare de șir de biți după configurarea inițială) și un mecanism de delegare pentru chei secrete.

Dovada securității pentru transformarea CCA

Dovedim mai întâi siguranța transformării noastre dintr-o schemă KEM într-o schemă CPA sigură. Am oferit o secvență de jocuri hibride, primul fiind securitatea IND-CPA pentru o schemă ABE. Amândoi aratăm că orice atacator de timp polinomial probabilistic (PPT) are diferență neglijabilă de avantaj între jocurile consecutive și că avantajul oricărui atacator în ultimul joc este 0. Prin urmare, orice adversar PPT are cel mult un avantaj neglijabil. Începem prin a descrie secvența jocurilor.

GameCPA

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK, atacatorului.
2. Atacatorul trimite două mesaje M_0, M_1 către provocator.
3. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$
4. Provocatorul execută $\text{EncryptCPA KEM}(\text{PK}, A; u) \rightarrow (K', \text{CT} * 0)$ apoi setează $K = K'$.
5. Provocatorul calculează $G(K, l) \rightarrow y$
6. În cele din urmă calculează $\text{CT} * 1 = y \oplus M_\beta$
7. Textul cifrat al provocării $\text{CT} * = (\text{CT} * 0, \text{CT} * 1)$ este trimis atacatorului.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Game1

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK, atacatorului.
2. Atacatorul trimite două mesaje M_0, M_1 către provocator.
3. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$.
4. Provocatorul rulează $\text{EncryptCPA KEM}(\text{PK}, A; u) \rightarrow (K', \text{CT} * 0)$ apoi alege K aleatoriu din spațiul cheie KEM.
5. Provocatorul calculează $G(K, l) \rightarrow y$.
6. În cele din urmă calculează $\text{CT} * 1 = y \oplus M_\beta$.
7. Textul cifrat al provocării $\text{CT} * = (\text{CT} * 0, \text{CT} * 1)$ este trimis atacatorului.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Game2

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK, atacatorului.
2. Atacatorul trimite două mesaje M_0, M_1 către provocator.
3. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$.
4. Provocatorul rulează $\text{EncryptCPA KEM}(\text{PK}, A; u) \rightarrow (K', \text{CT} * 0)$ apoi alege K aleatoriu din spațiul cheie KEM.

5. Provocatorul alege $y \in \{0, 1\}$ uniform la întâmplare.
6. În cele din urmă calculează $CT * 1 = y \oplus M\beta$.
7. Textul cifrat al provocării $CT * = (CT * 0, CT * 1)$ este trimis atacatorului.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Lemă. Dacă ABE KEM-ul nostru de bază este sigur IND-CPA, atunci diferența de avantaj a oricărui atacator PPT în Game1 și GameCPA este neglijabilă.

Luați în considerare un algoritm B care joacă jocul de securitate ABE-KEM IND-CPA. Prima oară primește un PK publicitar de la provocatorul ABE și trece acest lucru la A pentru pasul 1. Apoi rulează singur pașii 2-3. În continuare, primește textul cifrat al provocării KEM $CT * 0$ și cheia K. În funcție de rotirea monezii provocatorului, KEM este fie ales aleatoriu, fie generat ca $\text{EncryptCPA KEM}(\text{PK}, A; u) \rightarrow (K', CT * 0)$. B execută pașii 5 - 8 folosind valoarea de provocare K dată de provocatorul PRG. Dacă atacatorul câștigă (adică $\beta = \beta'$) B produce 1 pentru a indica faptul că K a fost generat dintr-un apel de criptare; în caz contrar, rezultă 0 pentru a indica faptul că a fost ales aleatoriu.

Observăm că dacă K a fost generat din criptarea B simulează GameCPA și dacă a fost ales aleatoriu simulează Game1. Prin urmare, diferența de avantaj a oricărui atacator din Game1 față de GameCPA va fi avantajul lui B în jocul de securitate IND-CPA.

Lemă. Dacă generatorul nostru pseudo-aleatoriu G este sigur, atunci diferența de avantaj a oricărui atacator PPT din Game2 și Game1 este neglijabilă.

Luați în considerare un algoritm B care joacă jocul de securitate PRG. Este nevoie de o provocare $y \in \{0, 1\}$ uniform la întâmplare sau este generată ca $G(K, l) \rightarrow y$. Algoritmul de reducere execută pașii 1-4 din Game1 de unul singur, cu excepția faptului că nu alege K. Apoi rulează pașii 6 - 8 folosind valoarea de provocare y dată de provocatorul PRG. Dacă atacatorul câștigă (adică $\beta = \beta'$), ieșirile 1 B indică că y a fost ales în mod pseudorandom; în caz contrar, rezultă 0 pentru a indica că a fost ales aleatoriu.

Observăm că dacă y a fost ales pseudorandom, B simulează Game1 și dacă a fost ales aleatoriu simulează Game2. Prin urmare, diferența de avantaj a oricărui atacator din Game2 față de Game1 va fi avantajul B în jocul de securitate.

Lemă. Avantajul oricărui atacator (nu neapărat PPT) din Game2 este 0. Acest lucru rezultă din faptul că, din moment ce y este ales uniform la întâmplare, $y \oplus M_\beta$ va fi distribuit ca un șir uniform aleatoriu și nu va avea nicio informație despre β .

Teoremă. Presupunând că PRG subiacent este sigur și că CPA KEM subiacent este sigur, transformarea schemei de criptare a mesajelor ABECPA KEM către ABE produce și schema securizată IND-CPA.

Dovada transformării CCA

Acum dovedim siguranța transformării noastre dintr-o schemă ABE securizată IND-CPA într-un schemă CPA securizat CCA. Facem acest lucru oferind o secvență de jocuri hibride, primul fiind securitatea IND-CCA pentru o schemă ABAB. Aratăm că orice atacator de timp polinomial probabilistic (PPT) are diferență neglijabilă de avantaj între jocuri consecutive și că avantajul oricărui atacator în ultimul joc este 0. Prin urmare, orice adversar PPT are cel mult un avantaj neglijabil. Dovada noastră de securitate va modela o funcție hash H ca oracol aleatoriu.

Începem prin a descrie secvența jocurilor.

GameCCA

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK , atacatorului.
2. Provocatorul înregistrează toate interogările ($r \parallel K \parallel A$) și răspunde cu o ieșire aleatorie dacă aceasta este prima oară când a fost interogată intrarea. În caz contrar, redă răspunsul anterior. Această operațiune oracle se desfășoară pe tot parcursul jocului.
3. Provocatorul execută decriptarea oricărui CT de text cifrat trimis pentru orice număr polinomial de multe ori.
4. Provocatorul primește o structură de acces la provocare A^* de la atacator.

5. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$. Apoi calculează $\text{EncryptCCA KEM}(\text{PK}, A^*) \rightarrow (K', \text{CT}^*)$. Dacă $\beta = 0$ setează $K^* = K'$; altfel alege K la întâmplare.
6. Valorile CT^*, K^* sunt trimise atacatorului.
7. Provocatorul execută decriptarea pe orice text cifrat $\text{CT} \neq \text{CT}^*$ trimis pentru orice număr polinomial de ori.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Game1

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK , atacatorului.
2. Provocatorul înregistrează toate interogările $(r \parallel K \parallel A)$ și răspunde cu o ieșire aleatorie u dacă aceasta este prima dată când a fost interogată intrarea. În plus, calculează $\text{EncryptCPA0}(\text{PK}, A, M = (K, r); u) \rightarrow \text{CT}$ și introduceți ambele CT, u în tabel pentru $(r \parallel K \parallel A)$. (Doar u este dat înapoi ca răspuns.)
3. Când i se dă un CT cu text cifrat, contestatorul verifică dacă CT apare ca o intrare cu text cifrat în tabelul random oracle. Dacă da, se afișează valoarea K corespunzătoare în tabel; în caz contrar, rezultă \perp pentru a respinge.
4. Provocatorul primește o structură de acces la provocare A^* de la atacator.
5. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$. Apoi calculează $\text{EncryptCCA KEM}(\text{PK}, A) \rightarrow (K', \text{CT}^*)$. Dacă $\beta = 0$ setează $K^* = K'$; altfel alege K^* la întâmplare.
6. Valorile CT^*, K^* sunt trimise atacatorului.
7. Atunci când i se dă un text cifrat $\text{CT} \neq \text{CT}^*$, contestatorul verifică dacă CT apare în tabelul oracol aleatoriu. Dacă da, acesta afișează valoarea K corespunzătoare în tabel; în caz contrar, rezultă \perp a respinge.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Game2

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK , atacatorului.
2. Provocatorul înregistrează toate interogările $(r \parallel K \parallel A)$ și răspunde cu o ieșire aleatorie u dacă aceasta este prima dată când a fost interogată intrarea. În plus, calculează $\text{EncryptCPA0}(\text{PK}, A, M = (K, r); u) \rightarrow C$ și introduceți ambele C, u în tabel pentru $(r \parallel K \parallel A)$. (Doar u este dat înapoi ca răspuns.)

3. Când i se dă un CT cu text cifrat, contestatorul verifică dacă CT apare în tabelul oracle aleatoriu. Dacă da, se afișează valoarea K corespunzătoare din tabel; în caz contrar, rezultă \perp pentru a respinge.
4. Provocatorul primește o structură de acces la provocare A^* de la atacator.
5. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$. Apoi calculează textul cifrat al provocării alegând aleatoriu $K' \in \{0, 1\}^n$, aleatoriu $r \in \{0, 1\}^n$ și aleatoriu u . (Nota u nu este aleasă apelând oracolul ran-dom.) Apoi setează $\text{EncryptCPA0}(PK, A^*, M = (K', r); u) \rightarrow CT^*$. Dacă $\beta = 0$ setează $K^* = K'$; în caz contrar alege K^* la întâmplare.
6. Valorile CT^*, K^* sunt trimise atacatorului.
7. Când i se dă un text criptat $CT \neq CT^*$ contestatorul verifică dacă CT apare în tabelul oracle aleatoriu. În acest caz, scoate valoarea corespunzătoare K în tabel; în caz contrar, rezultă \perp a respinge.
8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Game3

1. Provocatorul execută configurarea și dă cheia publică a schemei ABE, PK, atacatorului.
2. Provocatorul înregistrează toate interogările ($r \parallel K \parallel A$) și răspunde cu o ieșire aleatorie u dacă aceasta este prima dată când a fost interogată intrarea. În plus, calculează $\text{EncryptCPA0}(PK, A, M = (K, r); u) \rightarrow C$ și introduceți ambele C, u în tabel pentru ($r \parallel K \parallel A$). (Doar u este dat înapoi ca răspuns.)
3. Atunci când i se dă un CT cu text cifrat, contestatorul verifică dacă CT apare ca un C în tabelul oracol aleatoriu. Dacă da, el afișează valoarea K corespunzătoare în tabel; în caz contrar, rezultă \perp pentru a respinge.
4. Provocatorul primește o structură de acces la provocare A^* de la atacator.
5. Provocatorul rotește apoi o monedă $\beta \in \{0, 1\}$. Apoi calculează textul cifrat al provocării alegând aleatoriu $K', \tilde{K} \in \{0, 1\}^n$, aleatoriu $r \in \{0, 1\}^n$ și aleatoriu u . (Nota u nu este aleasă apelând oracolul aleatoriu.) Apoi setează $\text{EncryptCPA0}(PK, A^*, M = (\tilde{K}, r); u) \rightarrow CT^*$. Dacă $\beta = 0$ setează $K^* = K'$; altfel alege K la întâmplare.
6. Valorile CT^*, K^* sunt trimise atacatorului.
7. Când i se dă un text cifrat $CT \neq CT^*$ contestatorul verifică dacă CT apare ca un C în tabelul randomoracle. Dacă da, se afișează valoarea K corespunzătoare în tabel; în caz contrar, rezultă \perp a respinge.

8. Atacatorul trimite $\beta' \in \{0, 1\}$ și câștigă dacă $\beta = \beta'$.

Lemă. Dacă schema de bază este securizată IND-CPA, atunci avantajul diferenței dintre GameCCA și Game1 al oricărui atacator PPT este neglijabil.

Singura diferență în cele două jocuri este modul în care sunt tratate interogările de decriptare. Luăm în considerare două cazuri. În primul caz, se dă o interogare CT unde CT este o intrare de text cifrat în tabelul oracole aleatoriu. Cu toate acestea, în acest caz CT este exact textul cifrat produs la criptare pentru a accesa structura A cu aleatoriile K, r. Deoarece este o criptare corectă a cheii K, decriptarea ar trebui să producă K. Prin urmare, decriptarea este corectă. În al doilea caz, CT nu este pe listă.

În acest caz, jocul original s-ar putea decripta, dar Game1 va respinge întotdeauna. Susținem că șansa ca jocul original să decripteze cu succes este neglijabilă. Luați în considerare un astfel de text criptat în care ExtractAccessStructure (PK, CT) = A.

Să presupunem acum că DecryptCPA' (SK, CT) = M' = (K', r') și u' = H (r || K || A). Algoritmul de decriptare CCA va respinge, cu excepția cazului în care EncryptCPA0 (PK, A, M' = (K', r'); u') → CT. Cu toate acestea, întrucât oracolul aleatoriu nu a fost interogat (r || K || A), probabilitatea ca acest eveniment să se întâmple este limitată de probabilitatea deschiderii unei ieșiri de text cifrat printr-o criptare pentru un mesaj dat (fără a cunoaște aleatoritatea utilizată pentru a cripta) .

Dacă schema de bază este securizată IND-CPA, aceasta trebuie să aibă loc cu o probabilitate neglijabilă. Deoarece probabilitatea de a produce un text cifrat care decriptează diferit este neglijabilă, diferența avantajului este neglijabilă.

Lemă. Dacă schema de bază este IND-CPA sigură, atunci avantajul diferenței în Game1 și Game2 al oricărui atacator PPT este neglijabil.

Fie (A*, K', r) tupla folosită pentru a crea textul cifrat de provocare. Fie ca EVENT să fie evenimentul în care atacatorul interoghează oracolul aleatoriu de pe această tuplă. Observăm că punctul de vedere al atacatorului în jocurile Game1 și Game2 sunt informații teoretic identice până când se întâmplă acest eveniment. Astfel, putem argumenta că diferența de avantaj este neglijabil de aproape dacă EVENIMENTUL se întâmplă cu o probabilitate neglijabilă.

Acum argumentăm că EVENIMENTUL se produce într-adevăr cu o probabilitate neglijabilă. Să presupunem că a existat un atacator A care a declanșat EVENIMENT cu probabilitate non neglijabilă, atunci putem rupe securitatea CPA subiacentă.

Creăm un algoritm de reducere B care face următoarele. Rulează jocul de securitate ca în Game1 , dar unde primește cheia publică ABE PK de la provocatorul IND-CPA. Notă în Game1 poate răspunde cheilor de decriptare fără cheia secretă. Primește A^* și apoi trimite $(A^*, M_0 = (K', r))$ și $(A^*, M_1 = (K', \tilde{r}))$ la provocatorul IND-CPA pentru r, \tilde{r} ales aleatoriu și primește textul cifrat înapoi CT^* pe care îl folosește pentru a simula jocul.

Rulează simularea până când apare o interogare pentru $(A^*, (K', r))$ sau $(A^*, (K', \tilde{r}))$. Dacă este pentru primul ghicește 0; în caz contrar, presupune că 1. Dacă niciunul dintre ei nu este interogat, este nevoie de o presupunere aleatorie.

Putem vedea că, dacă textul cifrat al provocării ar fi o criptare a (K', r) , o interacțiune oracle pe $(A^*, (K', r))$ ar avea loc cu probabilitate ϵ și o interogare pe $(A^*, (K', \tilde{r}))$ ar avea loc cu o probabilitate neglijabilă, deoarece \tilde{r} are o lungime a parametrului de securitate. La fel, dacă textul cifrat de provocare ar fi o criptare a (K', \tilde{r}) interogare anoraclu pe $(A^*, (K', \tilde{r}))$ ar avea loc cu probabilitatea ϵ și ar avea loc o interogare pe $(A^*, (K', r))$ cu probabilitate neglijabilă. Rezultă că B sparge IND-CPA cu un avantaj neglijabil de aproape de ϵ .

Lemă. Dacă schema de bază este IND-CPA sigură, atunci avantajul diferenței din Game2 și Game3 al oricărui atacator PPT este neglijabil.

Proba de securitate se mapează imediat la un joc IND-CPA, deoarece (1) nu este utilizată nicio cheie secretă pentru decriptare și (2) aleatoritatea în crearea porțiunii IND-CPA a textului cifrat de provocare este aleasă cu adevărat la întâmplare.

Un algoritm de reducere B va rula pașii 1 - 4 de la sine și va trimite $K' \parallel r$ și un random $\sim K \parallel r$ ca mesaje alături de structura de acces A^* (dată de atacator) către provocatorul IND-CPA și va primi înapoi un text cifrat C . IND-CPA. Apoi va folosi asta pentru a juca restul Game2 . Imaginea atacatorului este aceeași ca în Game3 când $\sim K$ este criptat. Când K' este criptat de provocatorul IND-CPA, este același lucru cu avantajul atacatorului în Game2 .

Lemă. Dacă schema de bază este securizată IND-CPA, atunci avantajul din Game3 al oricărui atacator PPT este 0.

Avantajul atacatorului care face distincția între $K = K'$ și K ales aleatoriu este neglijabil atunci când $\sim K$ este criptat (în loc de K').

Teoremă. Presupunând că schema ABE subiacentă este sigură IND-CPA, atunci schema transformată este sigură CCA în modelul oracol aleatoriu.

Teorema rezultă imediat din lemele de mai sus.

Capitolul 2 Implementarea aplicației

2.1 Modul de funcționare al aplicației

Pentru demonstrarea utilizării în practică a schemei de criptare CP-ABE am ales să implementez o aplicație care gestionează informațiile medicale ale utilizatorilor, denumită Dosar Electronic de Sănătate. Această aplicație ar putea fi folositoare pentru centralizarea datelor medicale și criptarea lor bazată pe rolul fiecărui utilizator, care este stabilit pe baza atributelor pe care acesta le deține.

Atributele sunt reprezentate de șiruri de caractere și se pot personaliza pentru orice context în care este folosită CP-ABE. În acest caz, atributele au fost alese doar în scop experimental ("Medical", "Nurse", "Family", "Cardiology", "Pulmonology", "Epidemiology", "Surgery", "Endocrinology", "Gastroenterology", "FamilyMedicine", "General", "Internal").

Pentru implementarea aplicației am ales modelul de client/server. Serverul are rolul de Autoritate Centrală și are funcțiile de înregistrare al unui cont, autentificare al unui utilizator, generare de chei pentru utilizatori, criptare de date, decriptare de date, stocarea informațiilor într-o bază de date și trimiterea datelor la utilizatori.

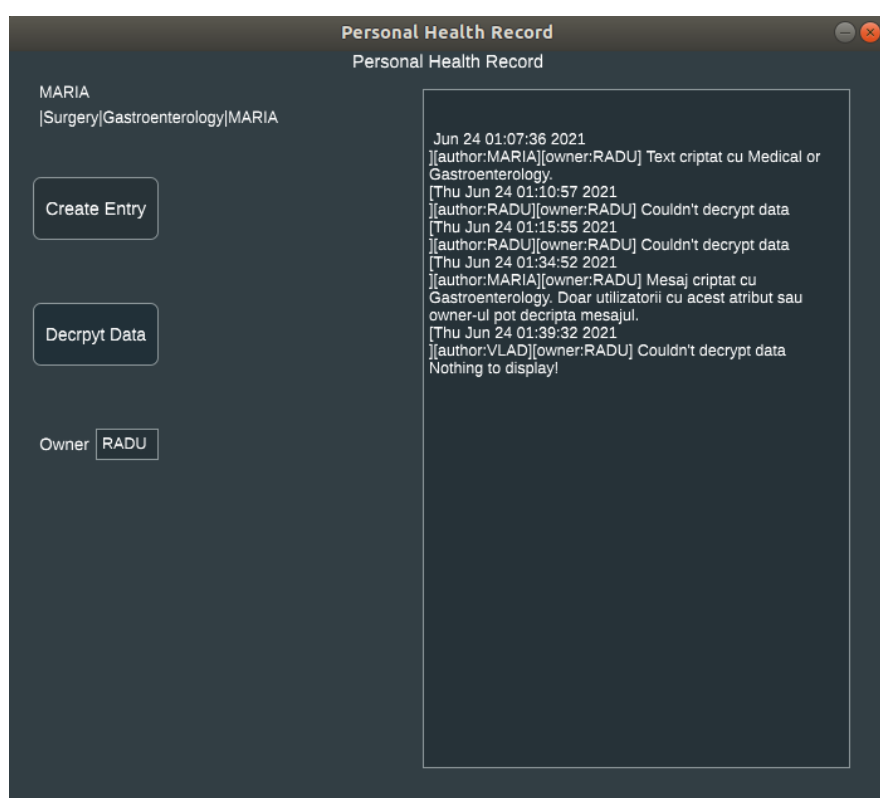
Clientul are rolul de a permite utilizatorilor să folosească funcțiile server-ului. La rândul lui are funcțiile de a afișa o interfață grafică ce este folosită pentru a prelua și trimite la server informațiile introduse de utilizator. De asemenea, prin intermediul clientului se pot afișa informațiile cerute (de exemplu dosarul personal al utilizatorului cu numele Sima Vlad).

FIGURĂ 1

La pornirea aplicației Client, pe ecran apare o fereastră cu două opțiuni, Login și Register. (vezi Figura 1). Utilizatorul poate crea un cont nou, sau se poate autentifica cu contul său. La crearea unui nou cont, serverul generează o cheie privată pentru acel utilizator nou, și trimite cheia utilizatorilor. La crearea cheii, pe lângă atributele acordate de Autoritatea Centrală se adaugă un atribut nou și unic pentru acel utilizator, care se folosește pentru derivarea cheii. În mod normal, Autoritatea Centrală este cea care are dreptul de a acorda atribute, și acest lucru este decis de organizația care folosește aplicația. În contextul acestei lucrări, am ales atribute aleatoare pentru utilizatori, pentru a demonstra funcționalitatea. Rolul atributului adițional este de a îmbunătăți granulara controlului accesului în modul următor. Când un mesaj este criptat, la regula de criptare a acelui mesaj se adaugă atributul unic al deținătorului mesajului. Astfel, un utilizator poate decripta toate mesajele din dosarul lui personal, chiar dacă cheia lui nu satisface regula de criptare (nu deține atributele necesare decriptării). De exemplu, un medic adăugat în dosarul personal al unui utilizator un mesaj criptat cu regula (Medical and Cardiology). Automat, regula de criptare este derivată în (Medical and Cardiology) or

AtributUnic. În cadrul aplicației implementată, atributul unic reprezintă numele de utilizator folosit la crearea contului. Acest lucru poate crea probleme în cazul în care contul unui utilizator este șters, apoi se creează un cont cu același nume, atunci utilizatorul care s-a înregistrat ulterior poate decripta mesajele criptate cu acel atribut. Însă acest lucru se poate remedia dacă se alege un atribut mult mai sigur (de exemplu CNP – ul unei persoane).

De asemenea, acest atribut mai poate fi folosit pentru delegarea unei singure persoane să decripteze un mesaj. Dacă un mesaj este criptat cu atributul unic al unui utilizator, atunci doar acel utilizator și deținătorul informației pot să decripteze acel mesaj. Dacă în contextul actual attributele unice sunt numele de utilizator, să considerăm ca VLAD criptează un mesaj cu regula (RADU). Prin modul de funcționare al aplicației, această regulă este derivată în (RADU or VLAD). Atunci doar acești doi utilizatori pot decripta mesajul.



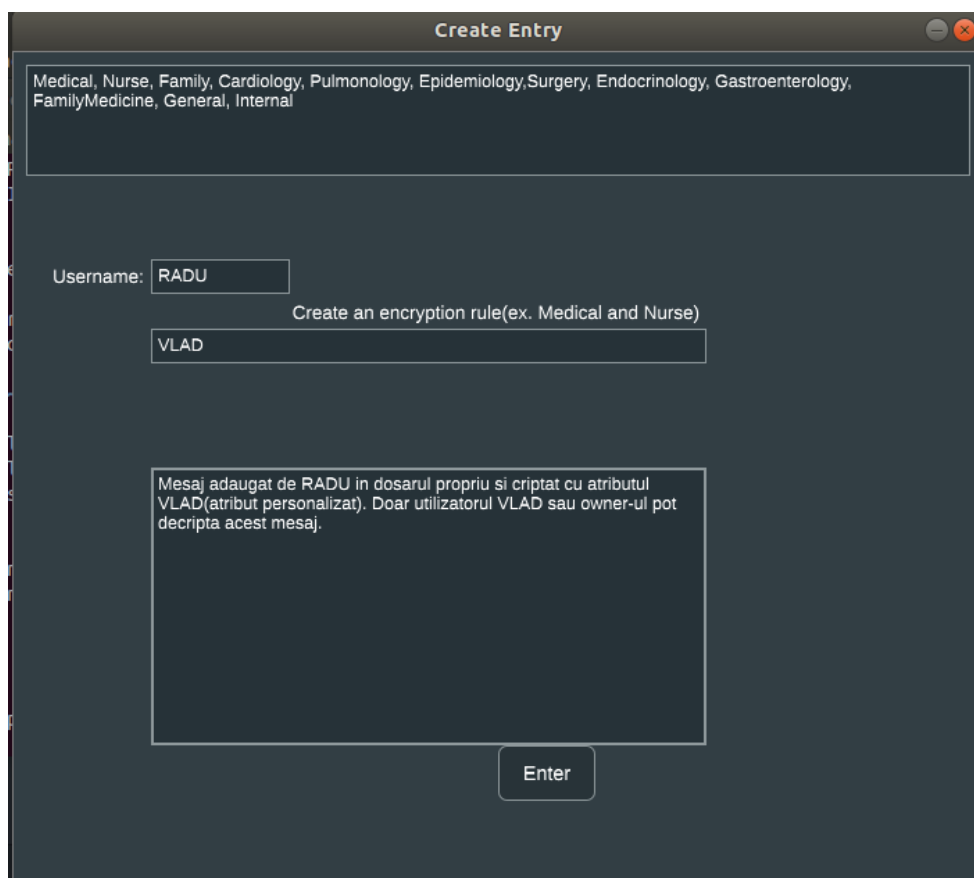
FIGURĂ 2

Dupa autentificare, un utilizator are opțiunea de a cere dosarul personal al unui utilizator. Acesta poate decripta informațiile doar dacă deține attributele cu care au fost criptat mesajele (vezi Figura 2).

De asemenea, Clientul are opțiunea de a adauga (vezi Figura 3) un mesaj în dosarul personal sau în dosarul altui utilizator, specificat în câmpul Username. Mulțimea de attribute disponibile este afișată în partea de sus a Clientului. Regula de criptare trebuie introdusă în locul specificat, și reprezintă o formulă logică formată din oricâte dintre attributele disponibile. De asemenea, mesajul care se dorește a fi criptat se introduce în locul specificat.

În imaginea din exemplu (Figura 3), utilizatorul RADU adaugă un mesaj în dosarul lui personal, criptat folosind atributul unic al utilizatorului VLAD.

Pentru a evita scrierea de mesaje de către utilizatori care nu ar trebui să aibă opțiunea asta, se poate crea un atribut care să specifice ca dacă un utilizator deține acest atribut, atunci el poate crea mesaje (de exemplu – atributul Medic).



FIGURĂ 3

Stocarea informațiilor se face într-o bază de date gestionată de server. Din moment ce în aceasta bază de date se stochează cheia master și datele criptate, este esențial ca accesul la această bază de date să fie restricționat și controlat cu strictețe. În acest context, serverul trebuie să fie de încredere și securizat.

În Figura 4 se poate vedea cum sunt stocate informațiile despre utilizatori în baza de date. Pentru o securitate mai mare a parolelor se poate utiliza o funcție hash (de ex. md5) pentru a cripta și stoca parolele în baza de date.

```
sqlite> select * from user;
1|VLAD|VLAD||Medical|Family|VLAD|AAAAF6pvyTmPkGIRCh+YnLF9VUVsM+9WTEFEAAABQqEBS6FES6FBAx/WynLR6mRLWEphP02zWPfWk0fXm+
HwwUdXclAxJcdjIP8pKzQRfMm9y+9/5p76NRarC3UIp9WjxhX86oP3wZyhCUTYX0ZhbwLseaEksqEhAg+hZv/wAvuXJzdDoc3Sdo4S+wMvX9GwOMPHC
NTBB54xoQpLWF9NZWRpY2FsoSSyoSEdHonFam3e03wpliQ0wzFWJt0+gWuE1rFIBIgNkEjuBQehB0tYX1ZMQUShJLKhIQMAH9lMEguCTZ+FgiTWBh70
Jo21j0Fp6UA0W8WEKseiqEBTKFES6FBAWzXxbzrSK9tPw+wFV0EoLdBy3ZX0WmsxP06E8iv5p0Clh2PSs4MpaerhChdQJ9p7j7THBvdgubTZRMuLw
lvFOhBWLuchV0oRV8TWkwnhBhXcYVW1pbhL8VxBRHw=
2|RADU|RADU||General|Internal|Endocrinology|RADU|AAAAF6pvyXWfS1nmHBqveVrxNb8bCdSQRVAAABjKEBS6FES6FBAiUEB8D9+skux7
HZMotuura5gurHtrM79L0xRvA0GyRGH8L9AAlLKoexRHLH67bbw13P/6Kh91jJi4FbCREybgGHEtYX0VuzG9jcmLub2xvZ3mJLKhIQMdoroaFg0mg
Fi6jbu/w6mhnrvm05FpdpPY3f2YzOfHA6EKS1hFR2VuzXJhbKEksqEhAwMzoJa8UM9j9FBu3Su7J0wq7bBVURMfgmEGG/RRH7xLoQtLWF9JbnRLcm5h
bKEksqEhAgbjttGh0ZmM0WzGxVFe6j6P/u7Zn42CEyGQMf9dNURaoQdLWF9SQRVVoSSyoSEdBRohrmBAQAvhgPvm2FB+c37iFeJbPFqzmqFmt7a+ka
hAUyhrLOhQQIMdHeI2m3qIhtUcgy1v4xn3BCCU2kGwdyeXzOVbHrx7X+/4wZVfNGOVamYqShWyKTnXDfD2HIYeY+ZzXcxJJoQVpbnB1dKELfEdlBm
VyyWx8SWS0ZJuvYx8RW5kb2Nyaw5vbcG9neXxSQURVFA==
3|MARIA|MARIA||Surgery|Gastroenterology|MARIA|AAAAGKpvyRmq8op0wHFQg/ePLfmdn3FNQVJJQQAaAViHAUuHRLohQQIO+RfPkbK2RLCTm
Z2o548xNBsu4KtgT7qoxk9/LG+jyD5Y2wwl0j0gDMhvGV+qSufOXuJKAz3lkg91AK5x02VoRNLWF9HYXN0cm9lbnRLcm9sb2d5oSSyoSEdGfFeu+Q
c7vkrYlZIL/FMY9R/SD8htd8RCGEcydumILOhCEtYX01BUkLBoSSyoSEDAjAGB/sye6y4W3f0U3szpMBawbRiuhK8xolasjjxgKhCkYXN1cmclnm
hJLKhIQIF08v7QmqjyOm/eqK5V74GYQ4ZxVCThInhL8ow+z0v0E8TKFES6FBAxSpwQULz+UxrbSniAELQcLf5yalZnDd09si/ai8WBAZ0US0NVmX
Oeu9gx7ZycXFtmgdQ/2GinXamclGcTj0hBWLuchV0oSB8U3Vy2ZVyeXxHYXN0cm9lbnRLcm9sb2d5fE1BUklbFA==
sqlite>
```

FIGURĂ 4

În Figura 5 se poate vedea cum sunt stocate mesaje criptate în baza de date.

```
1|MARIA|VLAD|AAABsKETqm/JkgJbnMY5df4A6D6MabNqwbIBmKEJ019TdXJnZJ35oSSyoSECLNVQDXZuTX0QbvuUwS+6ZI3+V7GZvsf0R2FFESrtrChBKnFVxBRKEksqEhAxZm8B/Z
3vWghsF5JtHl3Wh4Zg3W8ONcrxk707f/coToQZDcH3pbWWhJLKhIQIFBRx0y3G0tJLTZiBnpH/jXeSv600T3UpIgaehDU19qEJRf9tdXJnZJ35oSSyoSEdCQ2pHw/k75p1vA6/tih60Tr
ZVBWyc6kjdrrr3daa5Qgx/NprXNLSFPn2V2VsXZ/U/9ZCBHKSosLgF4r4ZYLsAeGRF9WTEFE0U5zoUEDIL+2RQbLEdZiR11qqluDvedIR+clNmDIRcGDHwgyaYU9Kmcagel+vi7LasYK
fKmoKHEF5KBV11ga9bb19UEpKED0VE0UdAAAQLCUI+JqJYtMUDzdTcdW5sVmFvv8bgPH/rf1Xdt5Tfbu4zPFmKfXr9QtVhkJJjGEmT9mZf8l0CPjyGxk+luhBnBvBGLjeaEWHQA
AABEOU3Yy2VyeSkgb3IgvK8RAAAAL2HE6oARpICW5zGOXX+A0g+jAGzasGhpqEQC1ShAR0AAABKILYQo/imugLn6mrc6HPqBPXj8oHm3KsfFgcpfbcbSiYX1LowE79SGkAgKBRJjroPa
l6Hjg/cDI2NgzWahYDooZ3QLZ3nTi4F00WFRk+2/OLu2F+H2ZcGUY8ERDoLhiDi6qPECSvAhFR0AAAQEW2rjTrl6etddzyiXXJtJqEDVGFnoRUDAAAAEH+RHXw6PwK4/tMldJvLnG4
=|Thu Jun 24 01:02:17 2021

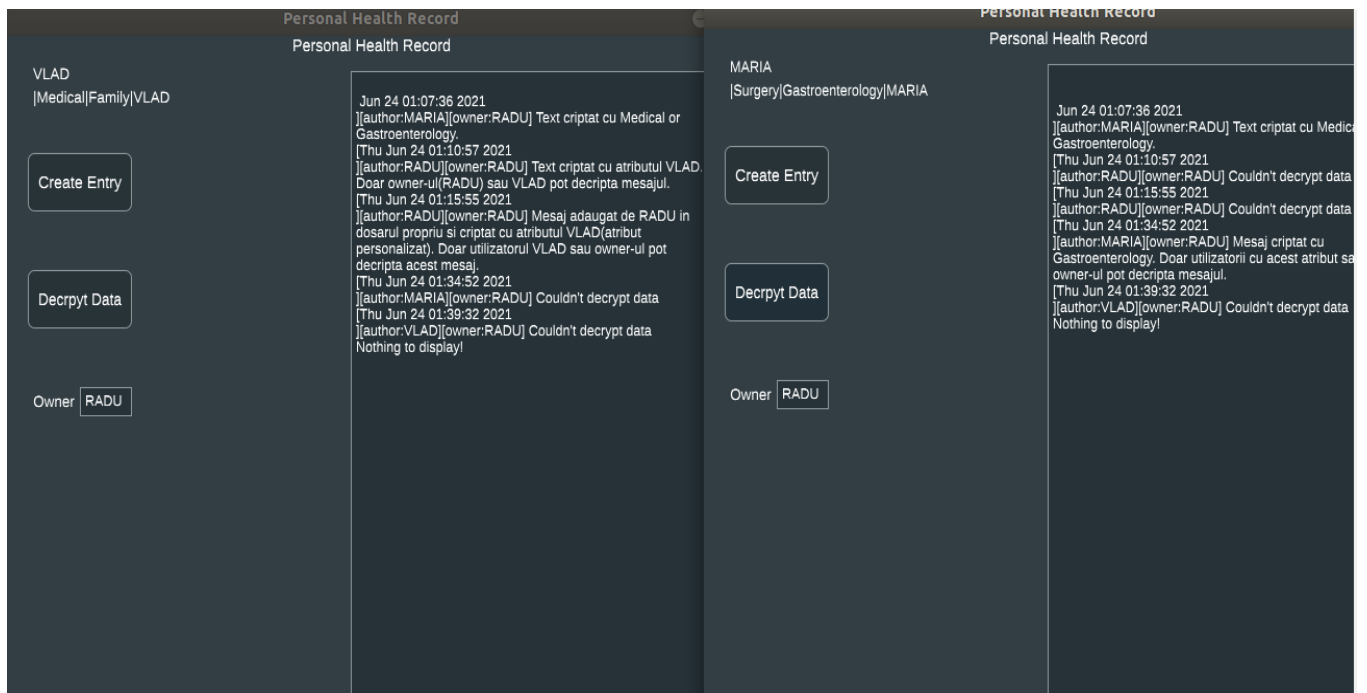
2|MARIA|RADU|AAACWKETqm/3eV0ltli7buMa6YqVQaPhH7ICQKESQ19HYXN0cm9lbnRLcm9sb2d5oSSyoSEDDunnPo8XSgCVfWtBvqH5Lk9EKoh8SvZBNmu94Y5vd0hCUNfTWwkaWNBh
KEksqEhAx2m0tDwJzuT0ZLjntWz6vn3o3muRkgc1GCMjKWNXjGoQZDX1JBRfWWhJLKhIQMH4/Pc9CEB5hgcFnaJLXbPoDRLI72M3yVnFclZQ7n6EGQ3BYaW1loSSyoSEdAGbAmJG4hg
cPuxBnLTueukG1JLMWJ8IqvJSEVmwCJCEKRFr2FzdHJvZ50ZJvBg9neaFES6FBAiPSLUvVwMFVw+Jf1rc84kjKAS1JPdvq+F4gWzRHsKvSbVrfghbe7D0a8j2GhXSRQdL/lpeKX+DGD
Bg9S6QDFDzShCURfTWwkaWNBhKFEs6FBAgMDmVHK5URdRD+pSVqEWIyFvCB06pTvdEoAvsckWkgHBE96YHzaVXSokNeEcCehRCKhXGgIrJldL06AvevJlHbKrfUkFEVAFes6FBAws8JSu
fNgthLLNDlIAIMV0aba0Zz/jzuVGD01ny1i8jz3B0HaKE+dGi2TSbquy1k82UgtHjgh8i1KUK7pCyXk19mha19FRKF9HQAAAEDK0XfQvFigtW2dr9YBhzQ7mLVHDQJJaRa0bVD/UVZoTqIY
ECza73XUWYwXlFA+AVJH2+VF4br/0jfcFamZeYoQZwb2xpY3mhKh0AAAA1KE1LZGLjYwWgb3Igr2FzdHJvZ50ZJvBg9neaSkgb3IgvKFEVQAAAIWH6oARnldJbZyU27jGunKLUghoR+
hbqEQC1ShMR0AAAs6qCb1dPudTKnv8yQF59TtFxnN2EQddYLPJdwLq0ApIcpB6VYQd8iruNqKhAkLwoRUDAAAAEHAKep2IXDNDon5Sdmp6Q0hA1RhZ6EVHQAAABDLmQ/RpUAFHfbobz
8iXlZAw|Thu Jun 24 01:07:36 2021

3|RADU|RADU|AAABp6ETqm/JJfCPTAG0X+pgobLRsBiftLIBj6EQQ19SQRVVoSSyoSEDMJTLVXiPacYpRvKnL89Y9sS6ask4bC00uqOrj3Pj0hBKnFVxBRKEksqEhAx75NypwsCJpat
ghKMSa8uhrzYr/X5u6P4cme8ax0WdnoQZDcH3pbWWhJLKhIQIMbWfDI/VLNeUuuk2Lcn40TuvDBhTvkNuvKGYW98VpaEGRF9SQRVVoU5zoUEDGs/3sutBNRAilQrACB88ZqyehX2jB1HJ
CHNrt5rIT9oGoLFryLFH00nU4k4NIEwtIlgP3stI0mEH07FnsuFvg1aEGRF9WTEFE0U5zoUEDetIfmZTBR6edGwXJu725xkg9tIee4ahcVJBKnlLLMAImuX1FJdqEUG6cEnAKRVtKyb4J
iGLA9fcpMK0tdWaedX0VE0UdAAAQNVOCd2aH/ou8bWuLTi2E0eKlm/4if7GCLdw/1bSRdfBrIcapIJBwcpbu37oSOcANKgtH7x2kdGE0Zu4Qsgs2hBnBvBGLjeaETHQAAAA4oVkbX
RCkgb3IgvKFEVQAAAKuHE0ARoxXD0wBtF/qYKAS0bAyn7ShLKECQ1ShvX0AAABSciprEiKbcGTy6+u68DnPK49Hm0EYvzn0wBgFucqB5/CZ6ytjE2MVJiPSLrhM2P/QsIRqEF+Nd90S
XtIh1kQ1Wks+dGQ75dzYFtcrEFQhJG6CESvAhFR0AAAQnLFZVAORBa6sK+ZNQ8BR0aEDVGFnoRUDAAAAEAP4XlweVJfAel4/bolNto8=|Thu Jun 24 01:10:57 2021

4|RADU|RADU|AAABp6ETqm/JZ0+Bo9x/ecq70bgwY4/DQ7IBj6EQQ19SQRVVoSSyoSEDDV8Bj+Tz0eWp8f5f0d02aj7zoHu/5LrIfftiIng5l1ahBKnFVxBRKEksqEhAyTNCBROxyZWD
acFDLMwWnkSrccjdsXr6U99b/OHT7XoQZDcH3pbWWhJLKhIQMC30G5nHVcq+u4pgD53yUDEFevfJPK/ZShrSeEX8VL6EGRF9SQRVVoU5zoUECADNMVq6f15JUPXJ2HVeTCQVAlcTtQEV
Hh3R0ZYbD6gLDfZgG1Tc91qheX5BTpw9IHKpxc9qp6m5ISM55/v2KEGRF9WTEFE0U5zoUEDfXGJTL7LzH/ukrCmT55LC0KPGQuB9on7GMSExHucpgb1EGNkPePQYerAd8rJa5Fdb4HtL
k4aigD0VwKkmbAEDX0VE0UdAAAQJPqNRCZhnSx1JXI0xYHg0xqPk8G504HvH2D0emgglR8sXl7YrvURKTutNPSbB80opSNzY9pTbZowYvWZf/mkuhBnBvBGLjeaETHQAAAA4oVkbX
RCkgb3IgvKFEVQAAAPohE0ARmdPgaPcf3nKuw24MGOPw0h3KECQ1Shnx0AAACpNawU/cdyjmn8aunTKUMFH5c6ZbmnmnmTgkzsQ3bYQV08mhjMBfNMLYcr/Q9GUH1zWl7moXiId09r
```

FIGURĂ 5

În Figura 6 se poate observa funcționarea modificărilor schemei de criptare implementate pentru această aplicație. Utilizatorii VLAD și MARIA au încercat să decripteze dosarul utilizatorului RADU, însă au reușit să decripteze doar mesaje ale căror regulă de criptare este satisfăcută de cheile lor.



2.2 Tehnologii folosite

Pentru implementarea aplicației client/server am folosit limbajul de programare C/C++.

“C++ este un limbaj de programare general, compilat. Este un limbaj multi-paradigmă, cu verificarea statică a tipului variabilelor ce suportă programare procedurală, abstractizare a datelor, programare orientată pe obiecte. În anii 1990, C++ a devenit unul dintre cele mai populare limbaje de programare comerciale, rămânând astfel până azi.”

Am ales acest limbaj de programare deoarece este performant și dispune de o librărie de lucru cu socket-uri, necesare pentru a stabili conexiunea între server și client. De asemenea, librăria ABE folosită este implementată folosind C++.

Server-ul implementat este un server TCP concurent multithreading. Un nou thread de execuție este creat pentru fiecare client care se conectează la server.

Pentru implementarea clientului am folosit framework-ul JUCE pentru interfețe grafice pentru utilizator (GUI).

Pentru gestiunea bazei de date am folosit SQLite, un sistem de gestiune a bazelor de date relaționale conținut într-o bibliotecă C. SQLite este rapid, ușor de utilizat și sigur.

Funcțiile criptografice folosite pentru implementarea schemei de criptare sunt din biblioteca OpenABE. OpenABE este o bibliotecă criptografică care încorporează o varietate de algoritmi de criptare bazată pe attribute (ABE), funcții și instrumente criptografice standard din industrie și o interfață intuitivă de programare a aplicațiilor (API). OpenABE este destinat să permită dezvoltatorilor să integreze fără probleme tehnologia ABE în aplicații care ar beneficia de ABE pentru a proteja și controla accesul la date sensibile. OpenABE este conceput pentru a fi ușor de utilizat și nu necesită ca dezvoltatorii să fie experți în criptare.

Capitolul 3 Concluzii și direcții viitoare

În această lucrare am exemplificat utilizarea schemei de criptare CP-ABE și am arătat cum se poate utiliza în practică. Exemplul se poate generaliza pentru o varietate de aplicații. Criptarea bazată pe atribute este o arie interesantă în Securitatea Informației. În opinia mea, această metodă este un instrument foarte util pentru proiectarea aplicațiilor care au nevoie de un control al accesului granular.

O noțiune de noutate pentru cercetătorii aceste arii a Securității Informației este funcția de delegare. În schemele tradiționale de criptare bazată pe atribute, controlul accesului este restricționat la cel mult un grup de utilizatori care au același atribut. Implementând o funcție de delegare, se poate obține controlul accesului personalizat.

Bibliografie

- Apu Kapadia, P. T. (2007, February 28). *Attribute-Based Publishing with Hidden Credentials and Hidden Policies*. Preluat de pe <https://www.ndss-symposium.org/wp-content/uploads/2017/09/Attribute-Based-Publishing-with-Hidden-Credentials-and-Hidden-Policies-Apu-Kapadia.pdf>
- Benaloh J., L. J. (1990, December 1). *Generalized Secret Sharing and Monotone Functions*. Preluat de pe DOI: https://doi.org/10.1007/0-387-34799-2_3
- Blakley, G. R. (fără an). Safeguarding Cryptographic Keys. *Proceedings of American Federation of Information Processing Societies*, pg. 313-317.
- Canetti R., H. S. (2004). *Chosen-Ciphertext Security from Identity-Based Encryption*. Preluat de pe DOI: https://doi.org/10.1007/978-3-540-24676-3_13
- Chase, M. (2007). *Multi-authority Attribute Based Encryption*. Preluat de pe <https://link.springer.com/>: https://doi.org/10.1007/978-3-540-70936-7_28
- D. Boneh, M. F. (2003). *Identity-Based Encryption from the Weil Pairing*. Preluat de pe <https://crypto.stanford.edu/>: <https://crypto.stanford.edu/~dabo/papers/bfibe.pdf>
- Ernest F. Brickell, D. M. (1991 , January). *On the classification of ideal secret sharing schemes*. Preluat de pe DOI: <https://doi.org/10.1007/BF00196772>
- Goyal V., J. A. (2008). *Bounded Ciphertext Policy Attribute Based Encryption*. Preluat de pe <https://link.springer.com/>: https://doi.org/10.1007/978-3-540-70583-3_47
- John Bethencourt, A. S. (2007). *Ciphertext-Policy Attribute-Based Encryption*. Preluat de pe <https://www.cs.utexas.edu/>: <https://www.cs.utexas.edu/~bwaters/publications/papers/cp-abe.pdf>
- Ling Cheung, C. N. (2007 , October). *Provably secure ciphertext policy ABE*. Preluat de pe DOI: <https://dl.acm.org/doi/10.1145/1315245.1315302>
- Matthew Pirretti, P. T. (2006, October). *Secure attribute-based systems*. Preluat de pe DOI: <https://dl.acm.org/doi/10.1145/1180405.1180419>
- Mitsuru Ito, A. S. (1989). *Secret sharing scheme realizing general access structure*. Preluat de pe DOI: <https://doi.org/10.1002/ecjc.4430720906>
- N. Saravana Kumar, G. R. (2014, December). *Enhanced Attribute Based Encryption for Cloud Computing*. Preluat de pe DOI: <https://doi.org/10.1016/j.procs.2015.02.127>.

- Robert W. Bradshaw, J. E. (2004). *Concealing complex policies with hidden credentials*.
 Preluat de pe <http://citeseerx.ist.psu.edu/>:
<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.88.267>
- Sahai A., W. B. (2005). *Fuzzy Identity-Based Encryption*. In: Cramer R. (eds) *Advances in Cryptology – EUROCRYPT 2005. EUROCRYPT 2005. Lecture Notes in Computer Science, vol 3494*. Springer, Berlin, Heidelberg. Preluat de pe <https://doi.org/>:
https://doi.org/10.1007/11426639_27
- Shamir, A. (1979, November 1). *How to share a secret*. Preluat de pe DOI:
<https://doi.org/10.1145/359168.359176>
- Smart, N. P. (2003, February 28). *Access Control Using Pairing Based Cryptography*. Preluat de pe link.springer.com: https://link.springer.com/chapter/10.1007%2F3-540-36563-X_8
- Vipul Goyal, O. P. (2006). *Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data*. Preluat de pe <https://eprint.iacr.org/>:
<https://eprint.iacr.org/2006/309.pdf>
- Waters, B. (2008). *Ciphertext-Policy Attribute-Based Encryption: An Expressive, Efficient, and Provably Secure Realization*. Preluat de pe <https://eprint.iacr.org/>:
<https://eprint.iacr.org/2008/290.pdf>