Cuprins

1 Introducere

Necesitatea de a cripta informaţiile a existat dintotdeauna. Astăzi, mai mult ca niciodată, informaţii confidenţiale se transmit zilnic între diverse entităţi. Pentru ca astfel de date sensibile să ajungă doar în mâna celor îndreptăţiţi să le afle - există criptografia.

În condițiile în care, la nivel global, doua din cinci copi ale fiecărui produs software distribuit sunt piratate conform studiului publicat de Global Software Survey în 2016 [19] companiile pierd anual circa $62.7 miliarde.

Este foarte important ca în sistemele de distribuit conținut atat la o scară larga cât și o scară mai mică anumite date să fie disponibile doar unui grup autorizat de utilizatorii. În distribuirea de conținut comercial, de exemplu, o companie ar putea dori ca informațiile sale să fie disponibile doar utilizatorilor care platesc.

O posibilă soluție la această problema poate fi folosirea criptării de tip broadcast. Totusi aceasta nu este o soluție optimă în toate cazurile deoarece de cele mai multe ori trebuie sa facem un compromis între o lungime foarte mare a criptotextului sau un numar mai mare de chei de criptare ce trebuie folosite în acest sistem.

Tema propusă în această lucrare, sistemele de criptare de tip broadcast, sunt o alternativă la criptarea cu chei publice iar acestea pot fi asociate diferitelor aplicații în care o entitate emite diferite diferite mesaje unui univers de utilizatorii și doar anumite grupuri privilegiate pot accesa mesajele.

În primă fază vom prezenta o construcție generică pentru sistemele de criptare de tip broadcast și diferite variații ale acesteia: sistmele BE bazate

pe identitate, sistemele BE bazate pe atribute și sistemele BE cu anonimat după care vom prezenta o comparatie între sistemele de criptare de tip broadcast și sistemele de criptare cu chei publice . În final, oferim câteva concluzii privind rezultatele din teză și identificăm unele probleme deschise în rezultatele noastre, dar și în literatura de specialitate pe care le sugerăm ca activitate viitoare de cercetare.

Aceasta lucrare este rezultatului efortului de selecție și sinteza dintre teorie și practica, al retinerii aspectelor esentiale în sistemele de criptare de tip broadcast constitue un punct de plecare pentru studiile viitoare în acest domeniu.

2 Sistem de criptare de tip Broadcast

Criptografia este știința scrierilor secrete. Ea stă la baza multor servicii și mecanisme de securitate folosite în internet, folosind metode matematice pentru transformarea datelor, în intenția de a ascunde continutul lor sau de a le proteja împotriva modificarii. Criptografia are o lunga istorie, confidentialitatea comunicarii fiind o cerință a tuturor timpurilor. Daca ar trebui sa alegem un singur exemplu al criptografiei 'clasice', acesta ar fi cifrul lui Cezar, nu atat datorita celebritatii imparatului roman de care se leagă folosirea lui, ci pentru că principiul sau de bază, al substituției, s-a mentinut nealterat aproape doua milenii.

În mod tradițional transmiterea securizată a informațiilor este realizată prin utilizarea sistemelor de criptare cu chei publice. Însă pentru ca acest sistem să funcționeze fiecare dispozitiv trebuie să cunoască anumite informații despre celelalte și să se pună de acord asupra cheiilor de criptare înainte de transmisie. Criptarea de tip broadcast rezolvă problema celor două dispozitive necunoscute anterior unul de altul prin stabilirea unei chei comune. Astfel se permite unor noi dispozitive, chiar dacă nu existau în momentul cand datele au fost criptate, să fie adăugate la un grup de dispozitive care pot decripta informația. Având în vedere că aceleași date sunt trimise tututor dispozitivelor în locul unui mesaj criptat pentru fiecare dispozitiv în parte, trebuie ca doar anumite dispozitive privilegiate să poată decripta mesajul.

Criptarea de tip broadcast este un tip de criptare propusă pentru prima dată de Fiat și Naor în anul 1993 în [1]. Scopul lor inițial a fost de a demonstra că două dispozitive care nu se cunoșteau anterior unul cu altul pot decide asupra unei chei comune pentru o comunicare sigură pe o cale one-way.

2.1 Schema generică

Un sistem de criptare de tip broadcast este construit din trei algoritmi randomizați:

- **Setup**(n): Primește la intrare numărul de receptori n și returnează cheile private $sk_1, ..., sk_n$ și o cheie publică PK.

- **Encrypt**(S, PK): Primește la intrare o submulțime de utilizatori $S \subseteq \{1, ..., n\}$, și o cheie publică PK. Returnează o pereche (Hdr,K) unde Hdr este numit antet iar $K \in \mathcal{K}$ este o cheie de criptare a mesajului aleasă din mulțimea finită de chei $\mathcal{K}$. Ne vom referi la Hdr ca criptotextul transmis.

- **Decrypt**$(S, i, k_i$,Hdr,$PK)$: Primește la intrare o submulțime de utilizatori $S \subseteq \{1, ..., n\}$ un identificator pentru un utilizator $i \in \{1, ..., n\}$, cheia privată sk_i pentru utilizatorul i, un antet Hdr, și cheia publică PK. Dacă $i \in S$, atunci algoritmul returnează o cheie de criptare a mesajului $K \in \mathcal{K}$ folosită de utilizator pentru a decripta mesajul.

Cerem ca sistemul sa fie corect pentru toate submulțimle $S \subseteq \{1, ..., n\}$ și toți $i \in S$,

dacă $Setup(n) \rightarrow (Pk, (sk_1, ..., sk_n))$ și $Encrypt(S, PK) \rightarrow$(Hdr,$K$) atunci $Decrypt(S, i$,Hdr,$PK)=K$.

Definim securitatea CCA a sistemului împotriva unui adversar static. Securitatea este definită folosind următorul joc între un adversar $\mathcal{A}$ și un challenger. Atât challengerul cât și $\mathcal{A}$ primesc n, numărul total de utilizatori, ca parametru de intrare.

- **Init**: Algoritmul $\mathcal{A}$ începe prin a returna o mulțime de receptori $S^* \subseteq \{1, ..., n\}$ pe care vrea să-i atace.

- **Setup**: Challengerul execută $Setup(n)$ pentru a obține cheia publică PK și cheile private $sk_1, ..., sk_n$ și trimite adversarului cheia publică PK și toate cheile private k_j pentru care $j \notin S^*$.

- **Etapa de interogare 1**: Adversarul $\mathcal{A}$ emite adaptiv interogări de decriptare $q_1, ..., q_m$ unde o interogare de decriptare constă în $(u, S$,Hdr) unde $S \subseteq S^*$ și $u \in S$. Challengerul răspunde cu $Decrypt$ (S, u, sk_u, Hdr, PK).

- **Provocarea**: Challengerul rulează algoritmul $Encrypt$ pentru a obține $(Hdr^*, K) \leftarrow Encrypt(S, PK)$ unde $K \in \mathcal{K}$. În continuare, challengerul generează aleatoriu un $b \in \{0, 1\}$. Setează $K_b = K$ și generează aleatoriu un $K_{1-b} \in \mathcal{K}$. Apoi trimite (Hdr*, K_0, K_1) adversarului $\mathcal{A}$.

- **Etapa de interogare 2**: Adversarul $\mathcal{A}$ emite adaptiv interogări de decriptare $q_{m+1}, ..., q_{q_D}$ unde $q_i = (u,\ S,\ \text{Hdr})$ cu $S \subseteq S^*$ și $u \in S$. Singura constrângere este ca Hdr $\neq$ Hdr*. Challengerul răspunde ca la pasul 1.

- **Presupunerea**: Adversarul $\mathcal{A}$ returnează o presupunere $b' \in \{0, 1\}$ pentru b și câștigă jocul dacă $b = b'$.

Fie $AdvBr_{\mathcal{A},n}$ probabilitatea ca $\mathcal{A}$ să câștige jocul atunci când challengerul primește n ca input.

Definiție 1. Spunem că un sistem de criptare de tip broadcast este CCA-sigur dacă pentru toți algoritmii A care emit q_D interogări de decriptare, avem $|AdvBr_{\mathcal{A}}, n - \frac{1}{2}| < \epsilon$.

Jocul de mai sus modelează un atac în care toți utilizatorii care nu sunt în mulțimea S^* complotează pentru a afla informații destinate doar celor din S^*. Mulțimea S^* este aleasă de adversarul $\mathcal{A}$. Trebuie menționat faptul că advarsarul este non-adaptiv; alege S^* și obține chei pentru utilizatorii din afara lui S^*, înainte de a vedea cheia publică PK. Un adversar adaptiv poate cere chei ale utilizatorilor în mod adaptiv.

Definiție 2. Spunem ca un sistem de criptare de tip broadcast este (t, ϵ, n)-sigur semantic dacă este $(t, \epsilon, n,0)$ CCA-sigur.

2.2 Funcții biliniare

Câteva informații necesare despre funcțiile biliniare și grupurile biliniare:

1. $\mathbb{G}$ și $\mathbb{G}_1$ sunt două grupuri multiplicative ciclice de ordin prim p.

2. g este generator al lui $\mathbb{G}$.

3. $e : \mathbb{G} \times \mathbb{G} \leftarrow \mathbb{G}_1$ este o funcție biliniară.

Fie $\mathbb{G}$ și $\mathbb{G}_1$ două grupuri ca mai sus. O funcție biliniară este o funcție $e : \mathbb{G} \times \mathbb{G} \leftarrow \mathbb{G}_1$ cu următoarele proprietăți:

1. Pentru orice $u, v \in \mathbb{G}$ și $a, b \in \mathbb{Z}$, avem $e(u^a, v^b) = e(u, v)^{ab}$

2. Funcția nu este degenerată, $e(g, g) \neq 1$.

Spunem că $\mathbb{G}$ este un grup biliniar dacă acțiunea sa poate fi calculată eficient și dacă există un grup $\mathbb{G}_1$ și o functie biliniară $e : \mathbb{G} \times \mathbb{G} \leftarrow \mathbb{G}_1$. Se observă faptul că e este simetrică: $e(g^a, g^b) = e(g, g)^{ab} = e(g^b, g^a)$.

2.3 Presupunerea de complexitate

Securitatea sistemului se bazează pe presupunerea de complexitate BDHE (Diffie-Hellman Exponent assumption):

Fie $\mathbb{G}$ un grup biliniar de ordin prim p. Problema ℓ-BDHE în $\mathbb{G}$ este următoarea: Dat un vector de $2\ell + 1$ elemente

$$(h, g, g^{\alpha}, g^{(\alpha^2)}, ..., g^{(\alpha^{\ell})}, g^{(\alpha^{\ell+2})}, ..., g^{(\alpha^{2\ell})})$$

să se returneze $e(g, h)^{(\alpha^{\ell+1})} \in \mathbb{G}_1$. Din vectorul de mai sus lipsește elementul $g^{(\alpha^{\ell+1})}$ deci funcția e este de ajutor în calcularea $e(g, h)^{(\alpha^{\ell+1})}$.

Folosim g_i pentru a nota $g_i = g^{(\alpha^i)} \in \mathbb{G}$.

Un algoritm $\mathcal{A}$ are avantajul ϵ în rezolvarea q-BDHE în $\mathbb{G}$ dacă

$$\Pr[\mathcal{A}(h, g, g_1, ..., g_{\ell}, g_{\ell+2}, ..., g_{2\ell}) = e(g_{\ell+1}, h)] \geq \epsilon$$

unde probabilitatea este peste generările aleatoarea ale generaratorului g în $\mathbb{G}$, generatorului h în $\mathbb{G}$, a lui α în $\mathbb{Z}_p$, și a biților folositi de $\mathcal{A}$.

Varianta decizională a problemei ℓ-BDHE în $\mathbb{G}$ este definită analog: Fie $y_{g,\alpha,\ell} = (g_1, ..., g_{\ell}, g_{\ell+2}, ...g_{2\ell})$. Un algoritm $\mathcal{B}$ care returnează $b \in \{0, 1\}$ are avantajul ϵ în rezolvarea problemei de decizie ℓ-BDHE în $\mathbb{G}$ dacă

$$\left| \Pr[\mathcal{B}(g, h, y_{g,\alpha,\ell}, e(g_{\ell+1}, h)) = 0] \text{ - } \Pr\ [\mathcal{B}(g, h, y_{g,\alpha,\ell}, T) = 0] \right| \geq \epsilon$$

unde probabilitatea este peste generările aleatoare ale generatorilor g, h în $\mathbb{G}$, a lui α în $\mathbb{Z}_p$, a lui $T \in \mathbb{G}_1$, și a bitilor folositi de $\mathcal{B}$. Ne vom referi la distribuția pe stânga prin $\mathcal{P}_{BDHE}$ iar la distributia pe dreapta prin $\mathcal{R}_{BDHE}$.

Definitie 3. Spunem că presupunerea de decizie (t, ϵ, ℓ)-BDHE se păstrează în $\mathbb{G}$ dacă niciun algoritm care se execută în timpul t nu are un avantaj mai mare de ϵ în rezolvarea problemei de decizie ℓ-BDHE în $\mathbb{G}$.

2.4 Schemă de criptare de tip broadcast rezistentă la coliziuni

2.4.1 Caz particular

Mai intâi vom discuta un caz special pentru n utilizatori în care dimensiunea criptotextului și a cheilor private este tot timpul constantă iar cheia publică creste liniar cu numărul de utilizatori.

- **Setup**(n): Fie $\mathbb{G}$ un grup biliniar de ordin prim p. Mai intâi generează aleatoriu un generator $g \in \mathbb{G}$ și $\alpha \in \mathbb{Z}_p$. Apoi calculează $g_i = g^{\alpha^i} \in \mathbb{G}$ pentru $i = 1, ..., n, n+2, ..., 2n$. Apoi, generează aleatoriu $\gamma \in \mathbb{Z}_p$ iar $v = g^\gamma \in \mathbb{G}$. Cheia publică este:

$$PK = (g, g_1, ..., g_n, g_{n+2}, ..., g_{2n}, v) \in \mathbb{G}^{2n+1}$$

  Cheia secretă pentru un utilizatorul $i \in \{1, ..., n\}$ este: $d_i = g_i^\gamma \in \mathbb{G}$. (Obs. $d_i = v^{(\alpha^i)}$). Returnează cheia publică PK și cele n chei secrete $sk_1, ..., sk_n$.

- **Encrypt**(S, PK): Generează aleatoriu $t \in \mathbb{Z}_p$ și setează $K = e(g_{n+1}, g)^t \in \mathbb{G}_1$. Valoarea lui $e(g_{n+1}, g)$ poate fi calculată ca $e(g_n, g_1)$. Apoi setează

$$\text{Hdr} = \left(g^t, (v \cdot \prod_{j \in S} g_{n+1-j})^t \right) \in \mathbb{G}^2$$

  și returnează perechea (Hdr,K).

- **Decrypt**$(S, i, sk_i$,Hdr$, Pk)$: Fie Hdr$=(C_0, C_1)$ iar $sk_i \in \mathbb{G}$. Atunci returnează:

$$K = e(g_i, C_1)/e(sk_i \cdot \prod_{j \in S, j \neq 1} g_{n+1-j+i}, C_0)$$

  O cheie secretă este doar un element al grupului $\mathbb{G}$ iar criptotextul, Hdr, doar două elemente. Deoarece $e(g_{n+1}, g)$ poate fi precalculat criptarea nu necesită cuplări. Cu toate acestea, sistemul este capabil să emită catre orice submulțime de utilizatori și este complet rezistent la coliziuni.

  Corectitudinea rezultă din faptul că algoritmul de decriptare funcționează corect. Se observă că pentru orice $i \in S$ coeficientul termenilor :
  $e(g_i, C_1) = e(g, g)^{\alpha^i \cdot t(\gamma + \Sigma_{j \in S} \alpha^{n+1-j})} = e(g, g)^{t(\gamma \alpha^i + \Sigma_{j \in S} \alpha^{n+1-j+i})}$ și
  $e(C_0, sk_i \prod_{j \in S, j \neq i} g_{n+1-j+i}) = e(g, g)^{t(\gamma \alpha^i + \Sigma_{j \neq i}^{j \in S} \alpha^{n+1-j+i})}$
  este $K = e(g_{n+1}, g)^t = e(g, g)^{t\alpha^{n+1}}$, după cum era cerut.

Pentru un număr foarte mare de receptori, timpul de decriptare va fi dat de cele $|S| - 2$ operații necesare pentru a calcula $\prod_{j \neq i}^{j \in S} g_{n+1-j+i}$. Însă, se obeservă ca daca receptorul a precalculat valoarea $w = \prod_{j \neq i}^{j \in S'} g_{n+1-j+i}$ pentru o mulțime de utilizatori S' care este similară cu S atunci, receptorul poate calcula $\prod_{j \neq i}^{j \in S} g_{n+1-j+i}$ cu doar δ operații folosind valoarea w, unde δ este dimensiunea mulțimii diferență dintre S și S'.

Această observație este utilă cand sistemul are de trimis către mulțimi mari de receptori, mulțimi de dimensiune $n - r$ pentru $r \ll n$. Cheia privată sk_i poate include valoarea $\prod_{j \neq i}^{j \in [1,n]} g_{n+1-j+i} \in \mathbb{G}$ ceea ce-i permite receptorului să decripteze folosind doar r operații iar utilizatorul i va avea nevoie de doar r elemente din cheia publică PK.

Timpul pentru criptare va fi dat de cele $|S| - 1$ operații necesare pentru a calcula $\prod_{j \in S} g^t_{n+1-j}$ cu o optimizare ca mai sus (precalcularea $\prod_{j=1}^{n} g_{n+1-j}$).

2.4.2 Caz general

În acest sistem idea este de a executa în paralel A instanțe ale sistemului descris mai sus unde fiecare instanță poate emite către cel mult $B < n$ utilizatori. Prin urmare sistemul se poate trata $n = AB$ utilizatori. Performanța este îmbunătățită prin împărțirea informației între cele A instanțe.Toate instanțele vor avea aceleași valori ale cheilor publice $g, g_1, ..., g_B, g_{B+2}, ..., g_{2B}$.

Dacă $B = n$ atunci acest sistem este cazul particular prezentat mai sus însă dacă $B = \lfloor \sqrt{n} \rfloor$ este un sistem în care atât dimensiunea cheii publice cât și a criptotextului este de aproximativ $\sqrt{n}$ elemente. Cheia privată este întotdeauna doar un element.

Fie B un număr întreg pozitiv fixat, atunci sistemul funcționează astfelȘ

- **Setup**(n): $A = \lceil \frac{n}{B} \rceil$ instanțe. Fie $\mathbb{G}$ un grup biliniar de ordin prim p. Mai întâi generează aleatoriu un generator $g \in \mathbb{G}$ și $\alpha \in \mathbb{Z}_p$. Calculează $g_i = g^{(\alpha^i)} \in \mathbb{G}$ pentru $i = 1, 2, ..., B, B + 2, ..., 2B$. Apoi generează aleatorriu $\gamma_1, ..., \gamma_A \in \mathbb{Z}_p$ și setează $v_1 = g^{\gamma_1}, ..., v_A = g^{\gamma_A} \in \mathbb{G}$. Cheia publică este:

  $$PK = (g, g_1, ..., g_B, g_{B+2}, ..., g_{2B}, v_1, ..., v_A) \in \mathbb{G}^{2B+A}$$

  Cheia privată pentru utilizatorul $i \in \{1, ..., n\}$ este definită după cum urmează: se scrie i ca $i = (a - 1)B + b$ pentru $1 \leq a \leq A$ și $1 \leq b \leq B$ ($a = \lceil i/B \rceil$ și $b = i \bmod B$). Cheia secretă pentru utilizatorul i este:

$$sk_i = g_b^{\gamma_\alpha} \qquad (sk_i = v_a^{(\alpha^b)})$$

Returnează cheia publică PK și n chei private $sk_1, ..., sk_n$.

- **Encrypt**(S, PK): Pentru fiecare $\ell = 1, ..., A$ definește submulțimile $\hat{S}_\ell$ și S_ℓ ca:

$$\hat{S}_\ell = S \cap \{(\ell-1)B+1, ..., \ell B\}, \quad S_\ell = \{x - \ell B + B \mid x \in \hat{S_\ell}\} \subseteq \{1, ..., B\}$$

Cu alte cuvinte, $S\hat{S}_\ell$ conține toți utilizatorii din S care sunt în al ℓ interval de lungime B iar S_ℓ conține indicii utilizatorilor comparativ cu inceputul intervalului. Generează aleatoriu $t \in \mathbb{Z}_p$ și setează $K = e(g_{B+1}, g)^{t \in \mathbb{G}_1}$. Setează

$$\text{Hdr} = \left(g^t, (v_1 \cdot \prod_{j \in S_1} g_{B+1-j})^t, ..., (v_A \cdot \prod_{j \in S_A} g_{B+1-j})^t \right) \in \mathbb{G}^{A+1}$$

Returnează perechea (Hdr,K) unde Hdr conține $A + 1$ elemente.

- **Decrypt**$(S, i, sk_i$,Hdr,$PK)$: Fie Hdr $= C_0, C_1, ..., C_A$ iar $sk_i \in \mathbb{G}$. Scrie i ca $i = (a-1)B + b$ pentru $1 \leq a \leq A$ și $1 \leq b \leq B$. Atunci

$$K = e(g_b, C_a) / e(sk_i \cdot \prod_{j \in S_a, j \neq b} g_{B+1-j+b}, C_0)$$

Verificarea corectitudinii decriptarii se face analog cu cea a sistemului precedent. Se observă ca atunci când $B = n$ atunci $A = 1$ și obținem sistemul prezentat mai inainte.

Eficiența. Dimensiunea cheii private va fi de doar un element. Criptotextul conține $A + 1$ elemente iar cheia publică $2B + A$ elemente. Alegerea lui B depinde de aplicație.În unele cazuri se dorește ca $B = n$ pentru a obține cea mai mică dimensiune pentru criptotext. În alte cazuri se dorește ca $B = \sqrt{n}$ pentru a micșora concatenarea criptotextului și a cheii publice.

Timpul de decriptare pentru un utilizator $i = (a-1)B + b$ va fi de cele $|S_a| - 2 < B$ operații.

3 Sisteme de criptare de tip broadcast bazate pe identitate

IBBE este un criptosistem cu chei publice în care pentru criptarea mesajului se folosește orice șir care identifică în mod unic un utilizator (adresa de e-mail, CNP etc.) receptorul ca cheie publică simplificând multe aplicații care folosesc criptarea cu chei publice deoarece nu mai este necesara distribuirea de certificate.

3.1 Schema IBBE generică

O schemă IBBE implică o autoritate: Generatorul de chei private $\mathcal{PKG}$. $\mathcal{PKG}$ oferă fiecărui utilizator nou (cu identitate ID_i) capacitatea de a decripta mesaje prin furnizarea unei chei de decriptare sk_{ID_i}. Generarea sk_{ID_i} este efectuată folosind o cheie master secretă MSK. Emițătorul criptează mesajele și le transmite grupurilor de utilizatori via canale broadcast. O schemă IBBE cu parametrul de securitate λ și cu dimensiunea maximă a grupului de receptori m este un tuplu de algortimi

$$IBBE = (Setup, Extract, Encrypt, Decrypt)$$

- **Setup**(λ, m): Primește la intrare parametrul de securitate și dimensiunea grupului de receptori m și returnează cheia publică PK și cheia master secretă MSK . MSK este trimisă la generatorul de chei secrete ($\mathcal{PKG}$) iar PK făcută publică.

- **Extract**(IMSK,ID_i): Primește la intrare cheia secretă master și identitatea utilizatorului ID . Extract generează o cheia privată pentru utilizator sk_{ID_i}.

- **Encrypt**(S,PK): Primește la intrare cheia publică și o mulțime de indentități $S = \{ID_1, ..., ID_s\}$ cu $s \leq m$ și returnează o pereche (Hdr,K) unde Hdr este numit antet iar $K \in \mathcal{K}$ unde $\mathcal{K}$ este mulțimea de chei pentru schema de criptare simetrică. Când un mesaj M$\in \{0,1\}^*$ este trimis utilizatorilor din S emițătorul generează (Hdr,K)$\leftarrow$ Encrypt(S,PK), calculează criptarea C_M a lui M folosind cheia simetrică $K \in \mathcal{K}$ și trimite (Hdr,S,C_M).

- **Decrypt**(S,ID,sk_{ID},Hdr,PK): Primește la intrare o submulțime $S = \{ID_1, ..., ID_s\}$ (cu $s \leq m$), un ID cheia privată corespunzătoare sk_{ID}, un antet Hdr și cheia publică PK. Dacă ID $\in S$, algoritmul returnează cheia de criptare K care este mai apoi folosită pentru a decripta (C_M) și recupera M.

Pentru m=1 IBBE devine clasicul IBE.

3.2 Modelul de securitate IBBE

Securitatea IND-sID-CCA a unui sistem IBBE.

Securitate sistemului este definită prin următorul joc jucat intre un adversar $\mathcal{A}$ și un challenger. Atât adversarul cât și challengerul au cunosc m, dimensiunea maximă a mulțimii de receptori S.

Init: Adversarul alege o multime $S^* = \{ID_1{}^*, ..., ID_s{}^*\}$ de indentități pe care vrea să le atace (cu $s \leq m$).

Setup: Challengerul rulează Setup pentru a obține cheia publică PK. El îi dă cheia publică PK lui $\mathcal{A}$.

Etapa de interogare 1: Adversarul $\mathcal{A}$ emite interogari $q_1, ..., q_{s_0}$ unde q_i este una din următoarele:

- Interogare de extracție (ID_i) cu constrângerea $ID_I \notin S^*$: Challengerul ruleaza Extract pe ID_i și trimite cheia privată rezultată adversarului.
- Interogare de decriptare, care constă în un triplu(ID_i, S, Hdr) cu $S \subseteq S^*$ și $ID_i \in S$. Adversarul răspunde cu Decrypt(S,ID_i, sk_{ID_i},Hdr,PK).

Provocarea: Când $\mathcal{A}$ decide că prima etapă este gata, challengerul rulează algoritmul Encrypt pentru a obține (Hdr^*,K) = Encrypt (S^*,PK) unde $K \in \mathcal{K}$ apoi selectează aleatoriu $b \leftarrow \{0, 1\}$, setează $K_b = K$ și K_{1-b} cu o valoare aleasă aleatoriu din K . Challengerul returnează(Hdr^*, K_0, K_1) la $\mathcal{A}$.

Etapa de interogare 2: Adversarul continua să emită interogări $q_1, ..., q_{s_0}$ unde q_i unde q_i este una din următoarele:

- Interogare de extracție (Id_i), ca în pasul 1.
- Interogare de decriptare, ca în pasul 1 , dar cu constrângerea Hdr $\neq Hdr^*$. Challengerul răspunde ca la pasul 1.

Presupunere: Adverarul $\mathcal{A}$ face o presupunere $b' \in \{0,1\}$ și căstigă jocul dacă $b = b'$.

Notăm prin q_D numărul total de interogări de decriptare și prin t numărul total de interogări de extragere care pot fi emise de adversar în timpul jocului. Fie t,m q_D parametrii atacului , notăm prin $Adv^{ind}{}_{IBBE}(t,m,q_D),\mathcal{A})$ avantajul pe ca care îl are $\mathcal{A}$ în câștigarea jocului:

$Adv^{ind}{}_{IBBE}(t,m,q_D),\mathcal{A}) = \left| 2 \text{ x } \Pr[\ b = b'] - 1| = |\Pr[b = b' \ |b = 1] - \Pr[b' = 1 \ |b= 0] \right|$

unde probabilitatea este luată peste alegerile aleatoare ale lui $\mathcal{A}$, chalengerul și toți algoritmi probabilisti rulați de chalenger.

Definitie 4. Fie $Adv^{ind}_{IBBE}(t,m,q_D,\mathcal{A}) = \max_{\mathcal{A}} Adv^{ind}_{IBBE}(t,m,q_D,\mathcal{A})$ unde maximul este luat peste toți algorimtii probabiliști $\mathcal{A}$ ce rulează în timp polinomial în raport cu λ. Spunem că o schemă de de criptare de tip broadcast bazată pe identitate IBBE este (t,m,q_D)-IND-sID-CCA sigură dacă $Adv^{ind}_{IBBE}(t,m,q_D,\mathcal{A})$ este neglijabil în raport cu λ.

Definitie 5. Spunem că o schemă de de criptare de tip broadcast bazată pe identitate IBBE este (t,m)-IND-sID-CPA sigură dacă este $(t,m,0)$-IND-sID-CCA sigură.(Există multe metode a transforma o schemă IND-ID-CPA într-o schemă IND-ID-CCA)

3.3 Aplicații

3.3.1 Rețele ad-hoc

În rețelele wireless ad-hoc, nodurile acționează ca routere cu IP-uri dinamice și au funcții de bază care redirecționarea pachetelor, rutarea și gestionarea rețelei. Într-o rețea deschisă toate dispozitivele pot învăța cum se efectează comunicarea, din acest motiv securitarea este esentială. Prin urmare un protocol de securitate este necesar pentru a proteja conținutul astfel încât doar utilizatorii din grupul ad hoc sa poată afla informația.

Y. Hu și N Mu propun un astfel de protocol în [7] care de fiecare dată când o nouă rețea ad hoc este construită protocolul cere fiecărui membru al rețelei să își transmită indentitatea pentru a construi cheia grupului.

Construcție

Fie $\bar{S} = (ID_1,ID_2,...,ID_m)$ mulțimea identităților membrilor grupului și presupunem ca fiecare membru a obținut parametrii publici (g,g_0,g_1,g_2,h,F) cu $g_1 = g^{\alpha}$ și $g_0 = g^r$ din PKG, unde g și g_2 sunt generatori ai lui $\mathbb{G}$ iar α și

r sunt selectați aleatoriu din $\mathbb{Z}_p^*$. F este o funcție hash rezistentă la coliziuni $(F : \mathbb{G}_1) \to \mathbb{Z}_p^*)$. În plus, cheia privată obținută din PKG a ficărui membru este:

$$d_{ID_i} = g_2^{\alpha}(g_1^{ID_1}h)^r$$

Aceste proceduri pot fi terminate în orice moment inainte ca rețeaua ad-hoc să se formeze.

Presupunem că o mulțime de membrii notată $S(S \subseteq \bar{S})$ vrea să formeze o rețea mobilă ad hoc. Protocolul decurge astfel:

- *Setup*: Dat parametrul de securitate descris mai sus, fiecare membru poate efectua următoarele calcule. Membrul selectează la început aleatoriu un $t \in Z_P$ și calculează $T_1 = G^t$, $T_2 = g_2^t$. Generalizând, setăm $S = (ID_1, ..., ID_s)$ pentru $s \leq m$. Apoi calculează
$$x_i = F(\hat{e}((g_1^{ID_i}h)^t, g_0)), i = 1, ..., s.$$
Folosind aceste x_i, membrul construiește
$$f(x) = \prod_{i=1}^{s}(x - x_i) = \sum_{i=0}^{s} a_i x^i,$$
unde a_i este coeficientul corespunzător lui x^i și $x \in Z_p$. Apoi calculează $(h_0, ..., h_s)$ astfel
$$h_0 = g^{a_0}, h_1 = g^{a_1}, ..., h_s = g^{a_s}$$
Prin urmare fiecare membru al lui S are o cheie de criptare $(h_0, ..., h_s, T_1, T_2)$. Acest tuplu nu se modifică decât atunci când un membru părăsește grupul sau noi membrii sunt adăugați.

- *Encrypt* : Fie K o cheie de sesiune. Un emițător din S efectuează următoarele: Selectează o cheie aleatoare $k \in Z_P$ și calculează
$$C_0 = Kh_0^k, C_1 = h_1^k, ..., C_s = h_s^k$$
În final, emițătorul produce criptotextul $Hdr = (C_0, C_1, ..., C_s)$ și îl trimite celorlalți membrii.

- *Decrypt* : Pentru a obține cheia K de criptare a mesajului încapsulată în antetul *Hdr*, utilizatorul cu identitatea ID_i și cheia privată corespunzătoare
$$d_{ID_i} = g_2^{\alpha}(g_1^{ID_1}h)^r$$
calculează (cu identitatea $ID_i \in S$ și o aplicație biliniară $\hat{e}$)
$$x_i = F(\frac{\hat{e}(d_{ID_i}, T_i)}{\hat{e}(g_1, T_2)}) \ ,$$

și

$$K = C_0 \prod_{j=1}^{s} C_j^{x_i^j}.$$

Corectitudine

Presupunem ca H dr este bine-creat pentru S. Obținem

$F(\hat{e}(d_{ID_i}, T_1)/\hat{e}(g_1, T_2))$

$= F(\hat{e}(g_2^{\alpha}(g_1^{ID_i}h)^r, g^t)/\hat{e}(g_1, g_2^t))$

$= F(\hat{e}((g_1^{ID_i}h)^r, g^t)\hat{e}(g_2^{\alpha}, g^t)/\hat{e}(g_1, g_2^t))$

$= F(\hat{e}((g_1^{ID_i}h)^r, g^t)\hat{e}(g_2^t, g^1)/\hat{e}(g_1, g_2^t))$

$= F(\hat{e}((g_1^{ID_i}h)^r, g^t))$

$= x_i$

și

$$C_0 \prod_{j=1}^{s} C_j^{x_i^j}$$

$$= K g^{ka_0} \prod_{j=1}^{s} g^{ka_i x_i^j}$$

$$= K g^{k \sum_{j=0}^{s} a_i x_i^j}$$

$= K(g^{f(x_i)})^k$

$= K.$

Obs. $f(x_i) = 0$, deci obținem $g^{f(x_i)} = 1$.

Eficiență

O rețea mobilă ad hoc este dinamică și prin urmare membrii lui S se modifică de fiecare dată când rețeaua se modifică. Acest protocol necesită doar adăugarea sau eliminarea *ID*-urilor membrilor în timpul executiei algoritmului *Setup* pentru a obține o noua cheie de criptare, ceea ce este mult mai eficient decat protocoalele precedente. În plus acest protocol este eficient din punct de vedere al calculelor deoarece criptarea și decriptarea necesită doar evaluări ale unei aplicații biliniare.

3.4 IBBE-fară oracole

În [8] este propusă un schemă eficientă IBBE fară oracole aleatoare care este IND-CPA sigură la identitate selectată.

3.4.1 Descrierea schemei

Schema IBBE fară oracole aleatoare bazată pe asumpția q-BDHI. Schema propusă oferă o dimensiune constantă a criptotextului, cheii publice și cheilor private.

Fie $\mathcal{G}$ un grup bilinear de ordin prim p. Alegem o funcție criptografică hash rezistentă la coliziuni $H : \{0,1\}^* \to \mathcal{Z}_p^*$, care atribuie arbitrar identități ca chei publice (ID) din $\{0,1\}^*$ în $\mathbb{Z}_p^*$. Presupunem că K este un element in $\mathcal{G}_1$, unde $K \in \mathcal{K}$ iar $\mathcal{K}$ este multimea de chei pentru schema de criptare simetrică.

- **Setup**(λ, m) :Pentru a genera parametrii sistemului IBBE primind paramatrul de securitate λ și un numar intreg m un sistem biliniar $\mathcal{G} = (p, \mathcal{G}, \mathcal{G}_1, e(\cdot,\cdot))$ este construit. Apoi generăm un generator aleatoriu $g \in \mathcal{G}^*$, doua elemente aleatoare $x, y \in \mathcal{Z}_p^*$, și calculăm $X = g^x, Y = g^y$. CHeia publică PK și cheia master secretă sunt definite după cum urmează:

$$\text{PK}=(g,X,Y),\ \text{MSK}=\ (x,y)$$

- **Extract**(MSK, ID_i) : Avem MSK=(x,y), pentru a creea cheia privată pentru cheia publică $ID_i \in \mathcal{Z}_p^*$:

  1. generăm aleatoriu un element $r \in \mathcal{Z}_p^*$, și calculăm $R = g^{\frac{1}{(r+ID_i)\cdot y+x}}$,
  2 . returnăm cheia privată $skID_i = (r, R)$

  În cazul în care $(r + ID_i) * y + x = 0$, trebuie să alegem o altă valoare pentru r.

- **Encrypt**(PK,$\mathcal{N}$,K): Notăm prin $\mathcal{N} = \{ID_j\}_{j=1}^n$ mulțimea receptorilor. Pentru a cripta o cheie $K \in \mathcal{K}$ a schemei de criptare simetrice, emițătorul trebuie sa aleagă aleatoriu $s \in \mathcal{Z}_p^*$, și să calculeze Hdr = (A,B,C,D) folosind PK și s pentru a încapsula cheia simetrică K, unde:

$$A = Y^{\prod_{j=1}^{n} ID_j \cdot s} \qquad B = X^s \qquad C = Y^s \qquad D = e(g,g)^s \cdot K$$

Trebuie menționat faptul că $e(g,g)$ poate fi precalculat .

- **Decrypt**(PK,$\mathcal{N}, ID_i, sk_{ID_i}$,Hdr): Pentru a recupera cheia de criptare k încapsulată în antetul Hdr = (A,B,C,D), receptorul din multimea $\mathcal{N} = \{ID_j\}_{j=1}^{n}$ cu identitatea $ID_i \in \mathcal{N}(1 \leq i \leq n)$ și cheia privată $sk_{ID_I} = (r, R)$ trebuie sa calculeze și să returneze $D/e(A^{1/(\prod_{j=1, j\neq i}^{n} ID_j)} \cdot B \cdot C^r, R)$. Întradevăr pentru criptotext valid avem:

$$\frac{D}{e(A^{\frac{1}{\prod_{j=1, j\neq i}^{n} ID_j}} \cdot B \cdot C^r, R)} = \frac{D}{e(g^{y \cdot ID_i \cdot s} \cdot g(x) \cdot s \cdot r, g^{\frac{1}{(r+ID_j)\cdot y \cdot x}})} = \frac{e(g,g)^s \cdot K}{e(g,g)^s} = K$$

3.4.2 Eficiența

Din punct de vedere al eficienței această schemă asigură $O(1)$ dimensiunea criptotextului ,a chei publice și a cheilor private. Aceasta rezultă din expresiile cheilor publice respectiv private după cum urmează:

- 1. Forma cheii publici este PK=(g, X, Y), unde $X = g^x, Y = g^y$ iar $x, y \in \mathcal{Z}_p^*$. Este evident faptul că cheia publică are dimensiune $O1$ și nu are nici o legatură cu numărul de receptori.

- 2. Forma cheilor private este $sk_{ID_I} = (r, R)$, unde $r \in \mathcal{Z}_p^*$ iar $R = g^{\frac{1}{(r+ID_i)\cdot y + x}}$. Este evident faptul că dimensiunea cheii private este constantă.

- 3. Forma criptotextului este Hdr = (A,B,C,D) unde :

$$A = Y^{\prod_{j=1}^{n} ID_j \cdot s} \qquad B = X^s \qquad C = Y^s \qquad D = e(g,g)^s \cdot K$$

unde $s \in \mathcal{Z}_p^*, K$ este cheia simetrică. Deoarece rezultatul lui $\prod_{j=1}^{n} ID_J \cdot s$ este o valoarea numerică, criptotextul nu are nici o legătură cu numarul de receptori și are dimensiunea $O(1)$.

3.4.3 Securitatea schemei

Această schemă este este IND-sID-CPA sigură sub presupunerea q-BDHI.

Teorema 1: *Presupunem că asumpția (t,q,ϵ) BDHI de decizie se păstrează în $\mathcal{G}$ de dimensiune $|\mathcal{G}| = p$. Atunci sistemul IBBE deinit mai înainte este (t',q_D, ϵ)-IND-sigur pentru orice $q_D < q$ și orice t' < t-$\Theta(\Gamma q^2)$ unde Γ este timpul maxim pentru o exponențiere în $\mathcal{G}$).*

Demonstrație: Presupunem că adversarul $\mathcal{A}$ are un avantaj ϵ în atacarea sistemului IBBE. Construim un algoritm $\mathcal{B}$ care folosește $\mathcal{A}$ pentru a rezolva problema de deicizie q-BDHI în $\mathcal{G}$. Algoritmul $\mathcal{B}$ primește ca intrare un $(q+2)$-tuplu $(g, g^{\alpha}, g^{\alpha^2}, ..., g^{\alpha^q}, T) \in (\mathcal{G}^*)^{q+1} \times \mathcal{G}_1$. Scopul algoritmului $\mathcal{B}$ este de a returna 1 dacă $T = e(g,g)^{1/\alpha}$ și 0 altfel. Algoritmul $\mathcal{B}$ funcționează interacționând cu $\mathcal{A}$ intr-un joc de selectare a indentității astfel:

- **Pregătire** : Algoritmul $\mathcal{B}$ crează un generator $h \in \mathcal{G}^*$ despre care știe $q-1$ perechi de forma $(w_i, h^{1/(\alpha \cdot w_i)})$ pentru $w_1, w_2, ..., w_{q-2} \in \mathcal{Z}_p^*$ aleatoare. Aceasta se face astfel:

  1. Generează aleatoriu $w_1, w_2, ..., w_{q-2} \in \mathcal{Z}_p^*$ și fie $f(z)$ polinomul $f(z) = \prod_{i=1}^{q-2}(z + wi)$. Dezvoltăm terminii lui f pentru a obține $f(z) = \sum_{i=0}^{q-2} c_i z^i$. Termenul constant c_0 este nenul.
  2. Calculează $h = \prod_{i=1}^{q-1}(g^{(\alpha)^i})^{c_i - 1} = g^{\alpha f(\alpha)}$ și $u = \prod_{i=1}^{q-1}(g^{(\alpha)^{i+1}})^{c_i - 1} = g^{\alpha^2 f(\alpha)}$. (Observatie $u = h^{\alpha}$).
  3. Verifică dacă $h \in \mathcal{G}^*$. Dacă $h = 1$ în $\mathcal{G}$ atunci $w_j = \alpha$ ceea înseamnă că $\mathcal{B}$ va putea rezolva provacarea direct în acest moment. De aceea presupunem că toti $w_i \neq -\alpha$.
  4. Se observă ca pentru orice $i = 1, ..., q-2$, este usor pentru $\mathcal{B}$, să construiască perechea $(w_i, h^{1/(\alpha \cdot w_i)})$ deoarece avem

     $$f_i(z) = \frac{\alpha f(z)}{\alpha w_i} = \frac{f(z)}{w_i} = \sum_{j=0}^{q-2} \frac{c_j}{w_i} z^j = \sum_{j=0}^{q-2} d_j z^j$$

     Apoi

$$h^{\frac{1}{\alpha \cdot w_i}} = g^{f_i(\alpha)} = \sum_{j=0}^{q-2} (g^{(}\alpha^j))^{d_j}$$

5. Mai departe, $\mathcal{B}$ calculează

$$T_h = T^{c_0^2} \cdot T_0$$

unde

$$T_0 = \prod_{i=0}^{q-2} \prod_{j=0}^{q-3} e(g^{(\alpha)^i}, g^{(\alpha)^j})^{c_i c_j + 1}$$

Se observă că dacă $T = e(g,g)^{1/\alpha}$ atunci $T_h = e(g^{f(\alpha)/\alpha}, g^{f(\alpha)}) = e(h,h)^{1/\alpha}$. Dimpotrivă dacă T este uniform în $\mathcal{G}_1$, atunci așa este si T_h.

Valorile h, u, T_h și perechile $(w_i, h^{1/(\alpha \cdot w_i)})$ pentru $i = 1, ..., q-1$ vor fi folosite de-a lungul simulării.

- **Initializare**: Jocul începe cu adversarul $\mathcal{A}$ returnând o mulțime $\mathcal{N}^* = \{ID_j^*\}_{j=1}^n$ de indentități pe care vrea să le atace.

- **Setup** Pentru a genera parametrii sistemului, algoritmul $\mathcal{B}$ face următoarele:

1. Generează aleatoriu $\alpha \in \mathcal{Z}_p^*$ și fie $b = \prod_{j=1}^{n} ID_j^*$.
2. Calculează $X = u^{a+b} = h^{\alpha(a+b)}$ și $Y = u = h^{\alpha}$
3. Publică PK=(h, X, Y) ca cheie publică. (Obs. X, Y sunt independente de ID_j^* din perspectiva adversarului.)
4. Definește implicit $x = \alpha(a+b)$ și $y = \alpha$, astfel încât $X = h^x$ și $Y = h^y$. Algoritmul $\mathcal{B}$ nu știe valorile lui x și y.

- **Etapa 1**: Adversarul $\mathcal{A}$ emite până la $q_D < q-1$ intergări despre cheia privată. Considerăm a i-a interogare pentru cheile private corespunzătoare cheii publice $ID_I \notin \{ID_j^*\}_{j=1}^n$. Trebuie să răspundem cu o cheie privată $(r, h^{\frac{1}{(r+ID_i)\cdot y + x}})$ pentru $r \in \mathcal{Z}_p$. Algoritmul $\mathcal{B}$ răspunde interogării dupa cum urmează:

1. Fie $(w_i, h^{1/(\alpha \cdot w_i)})$ a i-a pereche construită în etapa pregătirii. Fie $h_i = h^{1/(\alpha \cdot w_i)}$.
2. $\mathcal{B}$ construiește mai intâi un $r \in \mathcal{Z}_p$ ce satisface $(r + a + b) \cdot \alpha w_i = (r + ID_i) \cdot y + x$. Înlocuind valorile lui x și y ecuația devine:

$$(r + a + b) \cdot \alpha w_i = (r + ID_i) \cdot \alpha + \alpha(a + b)$$

Eliminând α din ecuație obținem $r = \frac{ID_i}{w_i - 1} - (a - b) \in \mathcal{Z}_p$ pe care $\mathcal{B}$ poate să-l evalueze.

3. Acum, $(r, h^{\frac{1}{r+a+b}})$ este o cheie privată validă pentru ID_i, pentru

$$h_i^{\frac{1}{r+a+b}} = (h^{\frac{1}{\alpha w_i}})^{r+a+b} = h^{\frac{1}{(r+ID_i) \cdot y + x}}$$

Din construcția lui r vedem că este uniform distribuit peste toate elementele din $\mathcal{Z}_p$ pentru care $(r + ID_i) \cdot y + x \neq 0$ și $r \neq -(a + b)$.

- **Provocarea**: Adversarul $\mathcal{A}$ returnează două mesaje $M_0, M_1 \in \mathcal{G}_1$, algoritmul $\mathcal{B}$ generează aleatoriu un bit $b \in \{0, 1\}$ și un $l \in \mathcal{Z}_p^*$. Răspunde cu un criptotext $CT = (h^{b \cdot l}, h^{(a+b) \cdot l}, h^l, T_h^l, M_b)$. Fie $s = l/\alpha$. Dacă $T_h = e(g, g)^{1/\alpha}$ avem:

$$h^{b \cdot l} = h^{\prod_{j=1}^{n} ID_j^* \cdot l} = h^{\alpha \cdot \prod_{j=1}^{n} ID_j^* \cdot s} = Y^{\prod_{j=1}^{n} ID_j^* \cdot s}$$
$$h^{(a+b) \cdot l} = h^{\alpha \cdot (a+b) \cdot s} = (h^x)^s = X^s$$
$$h^l = h^{\alpha \cdot s} = Y^s$$

Deci CT este o criptate validă a M_b pentru ID_i^*, cu $s = l/\alpha \in \mathcal{Z}_p$. Pe de altă parte dacă T_h este uniform în $\mathcal{G}_1$, din perspectiva adversarului,CT este independent de bitul b.

- **Etapa 2.**: Adversarul $\mathcal{A}$ emite mai multe interogări despre cheia privată, cel mult $q_D < q - 1$. Algoritmul $\mathcal{B}$ răspunde ca mai sus.

- Presupunere:În sfârșit, $\mathcal{S}$ returnează o presupunere $b' \in \{0, 1\}$. Dacă $b = b'$ atunci $\mathcal{B}$ returniază adică $T = e(g, g)^{1/\alpha}$ altfel returnează 0 ceea ce înseamnă că $T \neq e(g, g)^{1/\alpha}$.

4 Sisteme de criptare de tip broadcast bazată pe atribute

Criptarea bazată pe atribute este o abordare relativ recentă care duce mai departe conceptul de criptare bazată pe identitate în care cheia publică era un string arbitrar, de ex adresa de e-mail și defineste identitatea utilizatorilor ca o mulțime de atribute.

4.1 Politică de control a accesului

Pentru construirea unui sistem de criptare bazat pe atribute este necesară o politică de control a accesului. O politică de control a accesului este o politică care definește ce utilizatori au permisiunea de a accesa informații. De exemplu să considerăm scenariul următor: In mediu academic la catalogul cu note al elevilor au acces doar profesorii. Numeroasele acreditari ale profesorilor sunt numite atribute iar notiunea de profesor însăși este o politică de acces.

O politică de acces des folosită este cea booleană

Fie $\mathcal{U} = \{u_1, u_2, ..., u_\ell\}$ mulțimea tuturor utilizatorilor din sistem care pot să primească niște informații confindențiale. Pentru un utilizator $u \in \mathcal{U}$, $\mathfrak{B}(u)$ este multimea tuturor grupurilor de care aparține. De exemplu daca $\mathcal{U} = \{u_1, u_2, u_3\}$ și $\mathcal{G}_1 = \{u_1, u_2\}$, $\mathcal{G}_2 = \{u_2\}$ și $\mathcal{G}_3 = \{u_1, u_3\}$ atunci $\mathfrak{B}(u_1) = \mathcal{G}_1, \mathcal{G}_3$, $\mathfrak{B}(u_2) = \mathcal{G}_1, \mathcal{G}_2$ și $\mathfrak{B}(u_3) = \mathcal{G}_3$.

Vom nota prin A_i un atribut ce apartine unui utilizator din grupul $\mathcal{G}_i$ și prin $\bar{A}_i$ un atribut ce nu aparține unui utilizator din grupul $\mathcal{G}_i$. Grupurile de utilizatori pot fi organizate după o anumită proprietate pe care o au în comun. De exemplu localizarea lor geografică tipul de abonament sau orice altă proprietate. Vom nota prin $\mathcal{B}$ și $\bar{B}$ mulțimea de atribute pozitive $\mathcal{B} = \{A_1, ..., A_r\}$ și prin $\bar{\mathcal{B}} = \{A_{r+1}, ..., A_{r+s}\}$ mulțimea de atribute negative.

Conceptul de politică booleană de acces este central în criptarea de tip broadcast bazată pe atribute deoarece definește ce grupuri pot decripta sau nu un criptotext. De exemplu expresia $\mathbb{A} = \bar{A}_1 \wedge (A_2 \vee A_3)$ este o politică de acces booleană care permite tututor utilizatorilor din $\mathcal{G}_2$ sau $\mathcal{G}_3$ dar nu și din $\mathcal{G}_1$ să decripteze criptotextul. Expresiile booleene cele mai folosite în politicile de acces sunt cele în formă normală conjunctivă (CNF) scrisă ca $\bigwedge_{i=1}^{n} \bigvee_{j=1}^{m} a_{i,j}$ și cele în formă normală disjunctivă (DNF) scrise ca

$\bigvee_{i=1}^{n} \bigwedge_{j=1}^{m} a_{i,j}$ unde literalii $a_{i,j}$ pot fi negați sau nu. Aceste două forme sunt universale deoarece orice expresie booleană poate fi scrisă în oricare din cele două forme.

4.2 Schemă generică

O definiție formală pentru un sistem de criptare de tip broadcast bazat pe atribute constituit din trei algoritmi:

- **Setup**$(\lambda, \ell, \mathfrak{B}(u_i)_{1 \leq i \leq \ell})$: Acest algoritm primește la intrare parametrul de securitate λ, numărul total de utilizatori din sistem ℓ, mulțimea de atribute $\mathfrak{B}(u_i)$ pentru fiecare utilizator u_i. Returnează o cheie de criptare ek și ℓ chei de decriptare dk_i corespunzătoare fiecărui receptor.
- **Encrypt**$(ek, \mathbb{A})$: Acest algoritm primește la intrare cheia de criptare ek și politica de control a accesului $\mathbb{A}$ și returnează un antet Hdr precum și o cheie secretă SK$\in \mathcal{K}$ unde $\mathcal{K}$ este o mulțime finită de chei de criptare.
- **Decrypt**($\mathbb{A}$,Hdr,dk_i): Acest algoritm primește la intrare o cheie de decriptare dk_i, antetul Hdr și o politică de control a accesului $\mathbb{A}$ și returnează cheia secretă dk_i dacă $\mathfrak{B}(u_i)$ satisface $\mathbb{A}$ altfel returnează $\perp$.

Un astfel de sistem este corect pentru orice politică control a accesului $\mathbb{A}$ și orice mulțime posibilă de atribute $\mathfrak{B}(u_i)_{1 \leq i \leq \ell}$ dacă:

$(ek, dk_1, ..., dk_\ell = \textbf{Setup}(\lambda, \ell, \mathfrak{B}(u_i)_{1 \leq i \leq \ell})$ și (Hdr,SK) = **Encrypt**$(ek, \mathbb{A})$,
atunci **Decrypt**($\mathbb{A}$,Hdr,dk_i) = SK
pentru u_i-uri astfel încât $\mathfrak{B}(u_i)$ satisface $\mathbb{A}$ și
Decrypt($\mathbb{A}$,Hdr,dk_i) = $\perp$
pentru u_i-uri astfel încât $\mathfrak{B}(u_i)$ nu satisface $\mathbb{A}$.

Securitatea: Un sistem de criptare de tip broadcast bazat pe atribute este considerat sigur dacă atunci când cunoaște un antet și toate cheile de decriptare ale utilizatorilo revocați nu este posibil pentru un adversar să afle nici o informație despre cheia secretă. Să consideram următorul joc între un adversar $\mathcal{A}$ și un challenger $\mathcal{C}$.

1. Atăt challengerul cât și adversarul primesc un sistem de n atribute.

2. Adversarul $\mathcal{A}$ returnează o politică $\mathbb{A}$ cât și o mulțime de atribute $\mathfrak{B}(u_i)_{1\leq i\leq \ell}$ pe care intenționează să o atace.

3. Challengerul rulează algoritmul $Setup(\lambda, \ell, \mathfrak{B}(u_i)_{1\leq i\leq \ell})$ și îi trimite lui $\mathcal{A}$, cheia publică ek și cheia de decriptare dk_i corespunzătoare utilizatorului u_i pe care adversarul dorește sa-l atace.

3. Adversarul trimite două mesaje M_0, M_1 de lungime egală. Challegerul generează aleatoriu un bit $b \in \{0, 1\}$ pentru a cripta mesajul M_b. Challengerul rulează algoritmul $Encrypt(ek, \mathbb{A})$ pentru a obține Hdr și C pe care le trimite adversarului.

5. Adversarul $\mathcal{A}$ returnează presupunerea $b' \in \{0, 1\}$.

Adversarul câștigă jocul dacă $b = b'$ iar avantajul său este definit ca:

$$\mathrm{Adv}^{ind}(\lambda, n, \mathfrak{B}(u_i)_{1\leq i\leq \ell}, \mathcal{A} = |2\Pr[b = b'] - 1|)$$

unde probabilitate este luată peste bitul generat aleatoriu b și toți biții folosiți de algoritmii **Setup** și **Encrypt**.

Trebuie sa fie posibilă revocarea unui utilizator fără ai afecta pe utilizatorii nerevocați. De exemplu faptul ca un abonat la un anumit serviciu nu și-a plătit abonamentul nu ar trebui să afecteze alți abonați.

4.3 Sisteme ABBE pentru grupuri multi-privilegiate

Fie un grup cu multiple privilegii $SG_j 1 \leq j \leq m$, unde m este numarul de grupuri. Fie n_j numărul de utilizatori din SG_j și $ID_{i,j}, 1 \leq i \leq n_j$ identitatea utilizatorului al i-lea în grupul SG_j. Fie G, G_T două grupuri ciclice de ordin prim p și $\hat{e}$ o aplicație biliniară definită ca $\hat{e} : G \times G \to G_T$. Fie H_1 o funcție hash criptografică definită ca: $H_1 : \{0, 1\}^* \to G$ și P cheia master publică. Fie $\mathcal{N} = \{a_1, a_2, ..., a_t\}$ mulțimea de atribute. Considerăm o structură de acces ce constă doar în porti AND și care primește ca input literali notată ca $\wedge_{a_i \in I} \tilde{a_i}$, unde $I \subseteq \mathcal{N}$ și fiecare $\tilde{a_i}$ este un literal ($\tilde{a} = a_i$ *or not* (a_i)). În următoarea schemă propusă Setup este folosit pentru a genera parametrii publici iar Extract este folosit pentru a genera chei.

- **Setup**$(\lambda, \mathcal{N})$: Acest algoritm primește la intrare parametrul de securitate λ și mulțimea de atribute $\mathcal{N}$ și returnează parametrii publici ca

tuplu: Mai întâi alege o funcție hash criptografică $H_1 : \{0,1\}^* \to G$ și construiește aplicația biliniară $\hat{e} : G \times G \to G_T$. Apoi generează aleatoriu un întreg $pr \in \mathbb{Z}_p^*$ ca cheie master secretă. De asemenea generează parametrii publici $g^{h_i p}$ pentru atributele pozitive, $g^{h_i n}$ pentru atributele negative și $g^{h_i d}$ pentru celelalte. Returnează tuplul $< p, G, G_T, \hat{e}, P, H_1, \{g^{h_i p}\}_{1 \le i \le t}, \{g^{h_i n}\}_{1 \le i \le t}, \{g^{h_i d}\}_{1 \le i \le t} >$ ca parametru public.

- **Extract**(ID_{ij}, pr, S): Acest algoritm primeste la intrare indentitatea ID_{ij} a utilizatorului, cheia master secretă pr, și mulțimea de atribute S a utilizatorilor. Calculează cheia secretă $S_{ID_{ij}}$ corespunzătoare identității ID_{ij} ca $H_1(ID_{ij})^{pr}$. Calculează cheia pentru atributele pe care le are utilizatorul astfel: Pentru fiecare atribut din $\mathcal{N}$ generează aleatoriu $r_i \in \mathbb{Z}_p^*$ și calculează $\sum_{i=1}^{n} r_i = rpr$. Dacă $a_i \in S$ atunci $\mathcal{D}_i = (g^{r_i + h_i p})^{pr^{-1}}$ altfel $\mathcal{D}_i = (g^{r_i + h_i n})^{pr^{-1}}$. Pentru toți $a_i \in \mathcal{N}$, $F_i = (g^{r_i + h_i d})$. De asemenea $\hat{\mathcal{D}} = g^{-r}$ și returnează cheia secretă $SK =< S_{ID_{ij}}, \mathcal{D}_i, F_i, \hat{\mathcal{D}} >$.

- **Encrypt**$(M_{j1 \le j \le m}, W_j, S_j,)$:Acest algoritm este rulat de emitător pentru a obține criptotextul C_j corespunzător mesajului M_j pentru fiecare grup SG_j folosind cheia secretă corespunzătoare SK_j și T_i generat din politica de acces. K_{ij} este calculat pentru pentru fiecare utilizator din S_j folosind Y_1 unde Y_1 este calculat ca g^{y_1} cu un y_1 generat aleatoriu în $\mathbb{Z}_p^*$. Cel mai mic multiplu comun pentru toți K_{ij} este calculat ca Lcm_j pentru mulțimea S_j. Sistemul de congruențe, $X \equiv SK_j \ (mod\ Lcm_j)$ unde $SK_j = e(g,g)^{sk_j}, skj \in_R \mathbb{Z}_p^*$ este rezolvat folosind CRT pentru a obține X_j. Folosind politica W_j și $Y_2 = g^{y_2}$ unde $y_2 \in \mathbb{Z}_p^*$ efectuează următoarele: Pentru fiecare atribut pozitiv T_i este calculat ca $\hat{e}(g,g)^{(h_i p) y_2}$ iar pentru fiecare atribut negativ este calculat ca $\mathcal{D}_i = (g^{r_i + h_i n})^{pr^{-1}}$ iar pentru atributele rămase care nu sunt considerate în politică $\mathcal{D}_i = (g^{r_i + h_i d})^{pr^{-1}}$. Apoi calculează produsul T_i-urilor pentru a obține T_j iar mesajul este criptat ca $C_j = (T_j \oplus SK_j) M_j$. Emițătorul trimite $< Hdr, C >$ unde $Hdr =< X, W, Y_1, Y_2 >$, $X = \{X_j, 1 \le j \le m\}$, $W = \{W_j, 1 \le j \le m\}$ și $C = \{C_j, 1 \le j \le m\}$.

- **Decrypt**(SK, Hdr, C) :Acest algoritm calculează cheia K_{ij} cheia sa secretă $S_{ID_{ij}}$ și Y_1. El recuperează SK_j prin calcularea $X_j \ mod\ K_{ij}$. De asemenea calculează T_j folosind cheile sale pentru atribute.Pentru

atributele considerate pozitive în politică dacă are cheia secretă corespunzătoare $\mathcal{D}_i = (g^{r_i+h_ip})^{pr^{-1}}$, calculează $T_i = \hat{e}(\mathcal{D}_i, Y_2)$. Pentru atributele considerate negative în politică dacă are cheia secretă corespunzătoare $\mathcal{D}_i = (g^{r_i+h_in})^{pr^{-1}}$, calculează $T_i = \hat{e}(\mathcal{D}_i, Y_2)$. Pentru atributele rămase folosește $\mathcal{D}_i = (g^{r_i+h_id})^{pr^{-1}}$, și calculează $T_i = \hat{e}(\mathcal{D}_i, Y_2)$. Apoi calculează $\hat{e}(\hat{\mathcal{D}}_i, Y_2)$ și obține T_j. În final decriptează criptotextul C_j folosind SK_j și T_j pentru a obține mesajul M_j.

În acest sistem utilizatorii care au părăsit grupul nu mai cunosc cheile secrete folosite iar utilizatorii care au fost recent adăugați unui grup nu sut capabili să accese informații care au fost emise anterior. Criptarea se realizează folosind două chei Sk_j și T_j care sunt generate aleatoriu pentru fiecare sesiune. Doar după SK_j utilizatorii pot decripta folosind atributele ceea ce conferă forward și backward secrecy sistemului.

Pentru fiecare sesine se calculează valori noi pentru K_{ij} pentru toți utilizatorii selectați. Aceste K_{ij}-uri sunt determinate $y_1 \in \mathbb{Z}_p^*$ generat aleatoriu și de aceea Lcm, care depinde de K_{ij}, se schimbă de aseamenea pentru fiecare sesiune. De aceea sistemul de congruențe și soluția sa se schimbă în fiecare sesiune păstrânduse forward și backward secrecy. Deci un utilizator nou nu poate deduce chei vechi din moment ce nu a fost inclus în calcularea lui X_j. Similar un utilzator vechi care nu mai este un membru al grupului nu poate obține cheia SK_j din moment ce identitatea lui, ID_{ij}, nu a fost inclusă în calcularea lui SK_j.

Din moment ce cheile private ale utilizatorilor sunt independente de broadcast ei nu au o stare. Cheile private ale acestora nu se schimbă de la o sesiune la alta nu pot fi actualizate din moment ce utilizatorii nu au o stare.

De asemenea în acest sistem dimensiunea criptotextului este constantă deoarece emițătorul trimite $Hdr =< X, W, Y_1, Y_2 >$ unde $X = \{X_j, 1 \leq j \leq m\}$, $W = \{W_j, 1 \leq j \leq m\}$. Dimensiunea criptotextului este independentă de numărul de utilizatori și numărul de atribute. Sistemul asigură o dimensiune constantă a antetului Hdr pentru ficare grup constituit din patru elemente pentru decriptare.

5 Anonimitate

Este foarte important ca în sistemele de distribuit conținut atât la o scară largă cât și mică anumite date să fie disponibile doar unui grup autorizat de utilizatori. În distribuirea de conținut comercial, de exemplu, o companie ar putea dori ca informațiile sale sa fie disponibile doar pentru utilizatorii care plătesc. La o scară mai mică sa presupunem că departamentul unei facultăți trebuie sa acceseze lucrările de licență ale absolvenților. Dacă copiile electronice ale lucrărilor au fost stocate pe serverul facultății atunci ar trebui să fie accesibile doar membrilor facultății.

Adesea este la fel de importantă protejarea identității utilizatorilor care sunt capabili să acceseze conținut protejat. Site-urile comerciale de multe ori nu vor să dezvăluie indentitatea clienților deoarece concurenții ar putea folosi aceste date pentru face publicitate orientată spre acești clienți.

Metoda cea mai frecvent utilizată pentru a proteja atât conținutul electronic cât și intimitatea utilizatorilor care pot accesa este folosirea unui server de încredere. De fiecare dată când un utilizator dorește să acceseze conținut se autentifică pe server și îi este trimis conținutul pe un canal securizat. Atât timp cât serverul funcționează corect doar utilizatorii autorizați vor putea accesa conținutul iar informații privind identitatea lor nu vor fi divulgate, nici măcar altor utilizatori autorizați.

Deși această metodă simplă de protecție a datelor este adecvată pentru anumite aplicații are unele neajunsuri semnificative. În primul rând atât conținutul datelor cât și confindențialitate utilizatorilor este în pericol dacă serverul va fi compromis. În plus adesea furnizorii de conținut nu distribue conținutul în mod direct ci folosesc retele peer-to-peer. În acest caz furnizarea conținutului nu mai poate fi controlată în mod direct.

5.1 Sisteme broadcast cu anonimat

Definim o schemă de criptare de tip broadcast cu chei publice, unde algoritmii sunt construiți astfel încât să păstreze anonimitatea și sunt destul de generali încât să includă varianta de BE bazată pe identitate.

Fie $U = \{1, .., n\}$ universul utilizatorilor. O schemă de criptare de tip broadcast (BE) este definită prin patru algoritmi și are asociată spațiul mesajelor $\mathcal{MSP}$ și spațiul criptotextului $\mathcal{CSP}$.

- **Setup**(λ, n): Acest algoritm primește la intrare parametrul de securi-

tate λ și numărul de utilizatori din sistem n. Returnează o cheie master publică MPK și o cheie master secretă MSK.

- **Extract**(MPK, MSK, i): Acest algoritm primește la intrare cheia master publică MPK, cheia master secretă MSK și un index $i \in U$ și returnează cheia privată sk_i, pentru utilizatorul i.

- **Encrypt**(MPK, m, S): Acest algoritm primește la intrare cheia master privată MPK, un mesaj $m \in \mathcal{MSP}$ și o submultime de utilizatori $S \subseteq U$, către care va fi trimis mesajul. Returnează criptotextul $c \in CSP$.

- **Decrypt**(MPK, sk_i, c) Acest algoritm primește la intrare cheia master publică MPK, o cheie privată sk_i și criptotextul $c \in CSP$. Returnează fie mesajul $m \in \mathcal{MSP}$ fie simbolul $\perp$.

Pentru toti $S \subseteq U$ și $i \in U$, dacă $c = Encrypt(MPK, m, S)$ și sk_i este cheia privată pentru $i \in S$, atunci $Decrypt(MPK, sk_i, c = m)$ cu o probalitate mare.

Se observă ca această definitie nu mai necesită multimea S ca parametru pentru algoritmul de decriptare. Acest lucru este esențial pentru noțiunea de criptare broadcast anonimă (ANOBE) pentru care definim un model de securitate pentru cazul de adversari adaptivi.

Definiție 6. Definim securitatea sistemului la atacuri de tip criptotext ales$(ANO - IND - CCA)$ prin următorul joc.

- **Setup.** Challengerul $\mathcal{C}$ rulează $Setup(\lambda, n)$ pentru a genera perechea de chei master (MPK, MSK) și trimite MPK adversarului λ.

- **Pas 1.** $\mathcal{A}$ poate emite interogări către un oracol pentru orice index $i \in U$. Oracolul răspunde prin returnarea cheii private $sk_i = Extract(MPK, MSK, i)$. $\mathcal{A}$ poate de asemenea emite interogări de decriptare de forma (c, i), unde $i \in U$, iar oracolul returnează decriptarea $Decrypt(MPK, sk_i, c)$.

- **Provocarea.** $\mathcal{A}$ alege două mesaje de lungimi egale $m_0, m_1 \in \mathcal{MSP}$ și două mulțimi distincte $S_0, S_1 \subseteq U$ de utilizatori. Inpunem ca S_0 și S_1 să aibă același număr de elemente și restricția ca $\mathcal{A}$ să nu fi emis interogări pentru nici un $i \in S_0 \triangle S_1 = (S_0\ S_1) \cup (S_1\ S_0)$. În continuare, dacă există un $i \in S_0 \cap S_1$ pentru care $\mathcal{A}$ a interogat oracolul , cerem ca $m_0 = m_1$. Adversarul $\mathcal{A}$ trimite m_0, m_1 și S_0, S_1 la C. Cel din urmă alege aleatoriu

un bit $b \in \{0,1\}$ și calculează $c^* = Encrypt(MPK, m_b, S_b)$ pe care îl returneaza la $\mathcal{A}$.

- **Pas 2.** $\mathcal{A}$ continuă sa emită interogări catre oracol cu restrictiile $i \notin S_0 \triangle S_1$ și dacă $i \in S_0 \cup S_1$, atunci $m_0 = m_1$. $\mathcal{A}$ poate să continue să emită interogări de decriptare (c, i) cu restricția că dacă $c = c^*$ atunci fie $i \notin S_0 \triangle S_1$ fie $i \in S_0 \cup S_1$ și $m_0 = m_1$.

- **Presupunere.** Adversarul returnează presupunerea sa b' pentru b.

Definiție 7. Spunem că o schemă BE este anonimă și sigură semantic împotriva atacurilor de tip criptotext ales ($ANO - IND - CCA$) dacă toți adversarii $\mathcal{A}$ au un avantaj neglijabil în timpul jocului, unde avantajul adversarului este definit astfel

$$Adv_{\mathcal{A},BE}^{ANO-IND-CCA}(\lambda) = |Pr[b' = b] - \tfrac{1}{2}|$$

.

5.2 Construcții generice pentru ANOBE din PKE

5.2.1 ANOBE din presupuneri minimale

Prin simpla presupunere a existenței unei scheme PKE-IND-CCA sigură putem construi o schmeă ANOBE dupa cum urmează.

Fie $\pi^{pke} = (Gen, KeyGen, Encrypt, Decrypt)$ o schemă PKE cu spațiul mesajelor $\mathcal{M} = \{0,1\}^m$. Algoritmul Gen primește la intrare un parametru de securitate și returnează un parametru public par folosit de $KeyGen$ pentru a genera o pereche de chei (pk, sk). Fie $\sum = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ o schemă de semnatură ce constă într-un algoritm de generare $\mathcal{G}$, un algortim de semnare $\mathcal{S}$ și un algoritm de verificare $\mathcal{V}$. Presupunem că spațiul cheilor pentru $\sum$ este $\mathcal{K} = \{0,1\}^v$, pentru $v \in poly(\lambda)$. Folosim π^{pke} și $\sum$ pentru a instanția in mod general o schemă BE, cu spațiul mesajelor $\{0,1\}^{m-v}$. În continuare includem simbolul ϵ ca mesaj valid dar distins în $\{0,1\}^{m-v}$: cu alte cuvinte toate mesajele care sunt acceptate ca plaitext valid sunt diferite de ϵ.

- **Setup**(λ, n): Generează $\leftarrow Gen(\lambda)$ si pentru $i = 1, n$ genereză $(sk_i, pk_i \leftarrow KeyGen(par))$. Cheia master secretă este MSK=$\{sk_i\}_{i=1}^n$.

- **Key-Gen**(MPK,MSK,i): Analizează cheia secretă MSK ca $\{sk_i\}_{i=1}^n$ și returnează sk_i.

- **Encrypt**(MPK,M, S): Pentru a cripta M pentru o multime de receptori $S \subseteq \{1, ..., n\}$ genereaza o pereche de chei (SK,VK)$\leftarrow \mathcal{G}(\lambda)$. Pentru fiecare $j = 1, n$ calculează C_j=Encrypt$(par, pk_j, M \parallel VK)$ dacă $j \in S$ și C_j = Encrypt$(par, pk_j, \epsilon \parallel VK)$ dacă $j \notin S$. În final returnează $C = (C_1, ..., C_n, \sigma)$, unde $\sigma = S(\text{SK}(C_1, ..., C_n))$.

- **Dec**(MPK,sk_i, C): primeste criptotextul ANOBE $C = (C_1, ..., C_n, \sigma)$ și calculează M' =Decrypt(sk_i, C_i). Dacă $M' \neq \perp$, analizăm M' ca $M' = M \parallel VK$ pentru niște siruri de biți $M \in \{0, 1\}^{m-v}$. Apoi, dacă $\mathcal{V}$(VK $(C_1, ..., C_n), \sigma$)=1 și $M \neq \epsilon$ returnează M. Altfel returnează $\perp$.

Corectitudinea rezultă din corectitudinea π^{pke} și a $\sum$.

Teorema 2. Fie π^{pke} o schemă PKE IND-CCA sigură și fie $\sum$ o schemă de semnare one-time puternică. Schema BE construită mai sus este sigură impotriva adversarilor adaptivi.

Am descris o schemă ANOBE din presupuneri minimale. Trebuie menționat faptul că timpul de criptare este linear în n dar decriptare este efectuată în timp constant, deoarece un utilizator selectează criptotextul de decriptat in funcție de indexul său. Totuși, dimensiunea criptotextului este liniară în n din moment ce criptăm pentru fiecare utilizator.

Mai departe vom arătat cum să obținem o schemă ANOBE unde dimensiunea criptotextului este liniară în dimensiunea mulțimii $targetS$.

5.2.2 ANOBE bazat pe PKE

O soluție simplă este de a cripta mesajul cu cheia publică a fiecărui utilizator din multimea de utilizatori privilegiați. Această construcție naivă, diferă prin faptul că acum implementăm un o schemă de criptare cu chei publice doar pentru a cripta mesajul pentru utilizatorii din mulțimea țintă.

Pentru această abordare schema PKE trebuie sa fie cu chei private, unde criptotextul sa nu aibă scurgeri relativ la cheia publică cu care a fost creat. Se cere deasemenea ca schema PKE sa fie slab robustă, atât pentru corectitudine cât și pentru consistență în demonstrarea securității la un atac CCA. Mai departe vom demonstra ca este adaptiv sigur.

Fie $\pi^{pke} = (Gen, Keygen, Encrypt, Decrypt)$ o schemă PKE și fie $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ one-time sigmature. Schema noastră ANOBE, ANOBE$^{\pi^{pke}, \Sigma}$ este urmatoarea:

- **Setup**(λ, n): Rulează Gen(λ, n) pentru a obține parametrul public par. Pentru $i = 1$ pâna la n rulează Keygen(par) pentru a genera (sk_i, pki). Cheia master privată este MSK=$\{sk_i\}_{i=1}^n$ iar cheia master publică este MPK=$(par, \Sigma, \{pk_i\}_{i=1}^n)$.

- **Key-Gen**(MPK,MSK,i): Primește MSK=$\{sk_i\}_{i=1}^n$ și returnează sk_i.

- **Encrypt**(MPK,M, S): Pentru a cripta M pentru o multime de receptori $S = \{i_1, ..., i_\ell\} \subseteq \{1, ..., n\}$ de dimensiune $\ell = |S|$ genereaza o pereche de chei (SK,VK)$\leftarrow \mathcal{G}(\lambda)$. Pentru fiecare $j = 1, \ell$ calculează C_j= Encrypt$(par, pk_{ij}, M \parallel VK)$. Criptotextul ANOBE este C = (VK, $C_{\tau(1)}$,...,$C_{\tau(\ell)}$, σ), unde $\sigma = S$(SK,$C_{\tau(1)}, ..., C_{\tau(\ell)}$) iar $\tau : \{1, ..., \ell\} \leftarrow \{1, ..., \ell\}$ este o permutare aleatoare.

- **Decrypt**(MPK,sk_i, C): Analizează C ca tuplu (VK,$C_1, ..., C_\ell, \sigma$). Returnează $\perp$ dacă $\mathcal{V}$(VK,$C_1, ..., C_\ell, \sigma$) = 0. Altfel, se repetă acești pași pentru $j = 1$ pâna la ℓ.

  1. Se calculează $M' = Decrypt(sk_i, C_j)$. Dacă $M' \neq \perp$ și poate fi mai departe parsat ca $M' = M \parallel VK$ pentru M de lungime potrivită, returnează M.
  2. Dacă $j = \ell$ returnează $\perp$.

Corectitudinea ANOBE$^{\pi^{pke}, \Sigma}$ rezultă direct din corectitudinea și robustețea lui π^{pke}.

Teorema 3. ANOBE$^{\pi^{pke}, \Sigma}$ este ANO-IND-CCA adaptiv sigură presupunând că:

- (i) π^{pke} este cheie privată și sigură IND-CCA și robustă sub atacuri de tip criptotext ales

- (ii) Σ este schemă one-time signature greu de spart.

Din punct de vedere al eficienței, din această construcție vom obține scheme ANOBE care nu necesită foarte mult spatiu de stocare pentru pentru cheile private și criptotexte care sunt de $|S|$ ori dimensiunea criptotextului sub schema PKE. Criptarea și decriptarea au cost liniar în dimensiunea lui S.

5.2.3 Construcție generică pentru ANOBE din IBE

O schemă IBE-A constă în patru algoritmi (Setup,KeyExt,Enc,Dec), unde Setup și KeyExt sunt rulați de o autoritate de încredere (TA). Această construcție folosește o schema multi-TA I'= (CommonSetup, TASetup, KeyDer, Enc$'$, Dec$'$) descrisă în [10] și schema de semnare $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$. Vom folosi I' și Σ pentru a crea o schemă BE astfel.

- **Setup**(λ, n): Rulează CommonSetup cu parametrul de intrare $\lambda \in \mathbb{N}$ pentru a obține parametrul de sistem par. Rulează TASetup(par) de n ori pentru a obține n perechi distincte de chei master $\{mpk_i, msk_i\}_{i \in U}$. Returnează par, Σ și n chei publice $\{mpk_i\}_{i \in U}$.

- **Key-Gen**(par, λ, i): Returnează msk_i, cheia secretă corespunzătoare cheii publice mpk_i utilizatorului i.

- **Encrypt**(par, M, S): Rulează $\mathcal{G}$ pentru a obține o pereche de chei de semnare one-time (SK,VK). Pentru fiecare $i \in S$ rulează $Enc'(mpk_i$, M, VK) pentru a obține criptotextul C_i. Criptotextul ANOBE este C = (VK, $C_{\tau(1)}$,...,$C_{\tau(\ell)}$, σ), unde $\tau = S$(SK,$C_{\tau(1)}$,...,$C_{\tau(\ell)}$) și τ:$\{1, ..., \ell\}$ $\rightarrow \{1, ..., \ell\}$ este o permutare aleatoare.

- **Decrypt**(par, msk_i, C): Împarte C ca (VK, C_1,...,C_ℓ, σ). Dacă $\mathcal{V}$(VK,C_1, ..., C_ℓ, τ) = 0 returnează $\perp$. Altfel calculează ski_{VK} = KeyDer(mpk_i, msk_i, VK) și repetă următorii pași pentru $j = 1$ până la ℓ

  1 Calculează $M' = \text{Dec}'(mpk_i, ski_{VK}, C_j)$. Dacă $M' \neq 0$ și mai poate fi parsat ca $M' = M \parallel VK$ pentru M de lungime potrivită, returnează M.

  2. Dacă $j = \ell$ returnează $\perp$.

Corectitudinea schemei BE reiese în mod direct din corectitudinea și slab robustețea schemei I' folosită pentru a o construi. În această construcție cheile private au o dimensiune mică și constantă (doar un element din $\mathbb{Z}_p$) iar criptotextul cel mult $3 \cdot |S|$ plus o cheie de semnare și verificare. Cripatarea și decriptarea au ambele cost liniar în raport cu dimensiunea lui S.

Teorema 4. Fie I' o schema IBE TA-anonimă, sID-IND-CPA sigură și fie Σ semnătură one-time. Atunci schema BE de mai sus este adaptiv ANO-IND-CCA sigură.

5.3 Decriptare eficientă în modelul standard

Construcțiile generice pentru ANOBE prezentate mai sus suferă de decriptare în timp liniar. Aceasta rezultă din faptul ca utilizatoii nu știu care componentă a criptotextului este destinată lor, și prin urmare trebuie sa facă în medie $|S|/2$ decriptari înainte de a recupera mesajul. Evident această procedură este greoaie. Mai departe va fi prezentată o tehnică care atinge timp *constant* pentru decriptare în modelul standard. Vom folosi o noua primitivă numită sistem hint anonim bazat pe taguri.

5.3.1 Sistem Hint anonim bazat pe tag-uri

Un sistem hint anonim bazat pe tag-uri este un schema de criptare bazată pe tag-uri care permite generarea de forme slabe de criptare sub un tag t și o cheie publică pk. Rezultatul procesului constă într-o valoare U și un *hint* H. Perechea (U, H) trebuie să fie pseudo-aleatoare (în particular, hint-urile generate sub doua chei publice distincte trebuie să fie indistingibile) atunci când doar cheia publică pk este disponibilă. De asemenea, cheia privată sk face posibilă verificarea validității unui hint dat cu privire la un tag t. Formal un astfel de sistem este definit astfel.

- **Keygen**(cp): primește la intrare o mulțime de parametri publici comuni și returnează o pereche de chei (sk,pk). Presupunem ca cp specifică un spațiu neomogen $\mathcal{R}^h$ și un spațiu $\mathcal{T}^h$ de taguri acceptabile pentru sistem.

- **Hint**(cp,t, pk, r): este un algoritm determinist care primește la intrare parametrul public comun cp, o cheie publică pk, un tag t și un binar $r \in_R \mathcal{R}^h$. Returnează perechea (U, H) ce constă în o valoare U și un hint H. Trebuie ca U să depindă doar de r nu și de pk.

- **Invert**(cp,sk, t, U): este un algoritm determinist de "inversare" care primește la intrare o valoare U, un tag t și o cheie privată sk. Returnează fie un hint H fie $\perp$ dacă U nu este în mulțimea potrivită.

Pentru corectitudine cerem ca pentru orice pereche $(sk, pk) \leftarrow \text{Keygen}(\lambda)$ și orice r ales aleatoriu, dacă $(U, H) \leftarrow \text{Hint}(t, pk, r)$, atunci Invert(cp,$sk$, t, U) = H.

Definiție 4. Un sistem hint bazat pe tag-uri ($Keygen$, $Hint$, $Invert$) este anonim dacă niciun adversar PPT nu are un avantaj ne-neglijabil în următorul joc.

1. Pentru inputul parametrului comun cp, adversarul $\mathcal{A}$ alege un tag t^* și il trimite challengerului.

2. Challengerul generează două perechi de chei $(sk_0, pk_0) \leftarrow \text{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \text{Keygeb}(\lambda)$ și trimite pk_0, pk_1 la $\mathcal{A}$.

3. Polinomial în mai multe ocazii, $\mathcal{A}$ invocă adaptiv un oracol de verificare pe tripletul valoare-hint-tag (U, H, t) pentru ca $t \neq t^*$. Challengerul răspunde prin returnarea biților $(d_0, d_1) \in \{0,1\}^2$ unde $d_0 = 1$ dacă și numai dacă $H = \text{Invert}(\text{cp}, sk_0\ t, U)$ iar $d_1 = 1$ dacă și numai dacă $H = \text{Invert}(\text{cp}, sk_1\ t, U)$

4. Când $\mathcal{A}$ decide să intre în etapa provocării, challengerul dă cu banul un $b \leftarrow \{0,1\}$ și alege alte monede aleatoriu $r^* \leftarrow \mathcal{R}^h$. Returnează $(U^*, H^*) = \text{Hint}(\text{cp}, t^*, pk_b, r^*)$.

5. $\mathcal{A}$ poate face alte interogări care nu implică tagul țintă t^*.

6. $\mathcal{A}$ returnează un bit $b' \in \{0,1\}$ și câștigă dacă $b' = b$.

Ca de obicei, avantajul adversarului $\mathcal{A}$ este:

$$\mathbf{Adv}^{anon-hint}(\mathcal{A}) = |Pr[b' = b] - 1/2|$$

Definiție 5. Un sistem hint bazat pe tag-uri (Keygen,Hint,Invert) este puternic robust dacă niciun adversar PPT $\mathcal{A}$ nu are un avantaj ne-neglijabil în următorul joc, unde avantajul adversarului $\mathcal{A}$ este probabilitatea sa de a câștiga.

1. Challengerul alege parametrii publici cp și generează perechile de chei $(sk_0, pk_0) \leftarrow \text{Keygen}(\lambda)$, $(sk_1, pk_1) \leftarrow \text{Keygeb}(\lambda)$ și trimite cp, pk_0, pk_1 la $\mathcal{A}$.

2. $\mathcal{A}$ apelează oracolul de verificare pe triplete valoare-hint-tag (U, H, t) arbitrar alese. Challengerul răspunde prin returnarea biților $(d_0, d_1) \in \{0,1\}^2$ unde $d_0 = 1$ dacă și numai dacă $H = \text{Invert}(\text{cp}, sk_0\ t, U)$ iar $d_1 = 1$ dacă și numai dacă $H = \text{Invert}(\text{cp}, sk_1\ t, U)$.

3. $\mathcal{A}$ returnează un triplet (U^*, H^*, t^*) și câștigă dacă H^* = Invert(cp, sk_0, t^*, U^*) = 1 și H^* = Invert(cp, sk_1, t, U^*) = 1.

Slab robustețea pentru un sistem hint bazat pe tag-uri este definită prin lasarea adversarului să facă o provocare în pasul 3. Challengerul apoi alege un tag t^*, de asemenea alege aleatoriu o monedă r^*, generează o pereche valoare-hint (U^*, H^*) = Hint(cp, t^*, pk_0, r^*) și $\mathcal{A}$ câștigă dacă H^* = Invert(cp, sk_1, t^*, U^*) = 1.

Pentru a arăta că această primitivă este fezabilă vom da un exemplu de sistem hint anonim bazat pe presupunerea DDH și pe schema de scuritate CCA descrisă în [3].

Fie parametru public comun cp = $\{\mathbb{G}, p, g\}$ unde $\mathbb{G}$ este un grup de ordin prim $p > 2^\lambda$ cu generatorul $g \in_R \mathbb{G}$. Presupunem că tagurile sunt elemente ale $\mathcal{T}^h = \mathbb{Z}_p^*$ iar spațiul randomizat este $\mathcal{R}^h = \mathbb{Z}_p^*$.

- Keygen(cp): alege aleatoriu $x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}_p^*$ și calculează $X_i = g^{x_i}$ și $Y_i = g^{y_i}$ pentru fiecare $i \in \{1, 2\}$. Cheia publică este pk = (X_1, X_2, Y_1, Y_2) iar cheia privată este sk = (x_1, x_2, y_1, y_2).

- Hint(cp,t, pk, r): primește pk = ($\mathbb{G}$, p, g,X_1, X_2, Y_1, Y_2), returnează $\perp$ dacă $r \notin \mathcal{R}^h = \mathbb{Z}_p^*$. Altfel calculează (U, H) astfel:

$$U = g^r, \qquad H = (V, W) = ((X_1^t X_2)^r, (Y_1^t Y_2)^r).$$

- Invert(cp,sk, t, U): returnează $\perp$ dacă $U \notin \mathbb{G}$. Altfel, împarte cheia privată sk ca $(x_1, x_2, y_1, y_2) \in (\mathbb{Z}_p^*)^4$ și returnează $H = (V, W) = (U^{t \cdot x_1 + x_2}, U^{t \cdot y_1 + y_2})$.

5.3.2 ANOBE cu decriptare eficientă

Fie π^{hint} = (Keygen,Hint,Invert) un sistem hint anonim cu parametrul public comun cp. Fie π^{pke} = (Gen,Keygen,Encrypt,Decrypt) o schemă PKE și $\Sigma = (\mathcal{G}, \mathcal{S}, \mathcal{V})$ o schemă de semnare.

- Setup(λ, n): Obține (par) $\leftarrow$ Gen(λ) și pentru fiecare $i \in \{1, ..., n\}$ generează perechile de chei de criptare $(sk_i, pk_i) \leftarrow \pi^{pke}$Keygen($par$) și perechile de chei hint $(sk_i^h, pk_i^h) \leftarrow \pi^{hint}$Keygen(cp). Cheia master secretă este BE-MSK = $\{sk_i, sk_i^h\}_{i=1}^n$ iar cheia maste publică este BE-MPK = (cp, par, $\{(pk_i, pk_i^h)\}_{i=1}^n, \Sigma$).

- Key-Gen(BE-MPK,BE-MSK,i): împarte BE-MSK ca $\{sk_i, sk_i^h\}_{i=1}^n$ și returnează $sk_i = (sk_i, sk_i^h)$.

- Enc(BE-MPK,M, S): primind o mulțime de receptori $S = \{i_1, ..., i_\ell\} \subseteq \{1, ..., n\}$ de dimensiune $\ell = |S|$ și un mesaj M, generează o pereche de chei de semnare one-time (SK,VK) $\leftarrow \mathcal{G}(\lambda)$. Apoi alege aleatoriu o monedă $r \leftarrow \mathcal{R}^h$ pentru sistemul hint și calculează $(U, H_j) = \pi^{hint}$Hint(cp,VK,$pk_{i_j}^h, r$) pentru $j = 1$ pâna la ℓ (primul rezultat U al Hint nu depinde de cheia publică). Pentru $j = 1$ pâna la ℓ calculează $C_j = \pi^{pke}$Encrypt($par, pk_{i_j}, M||$VK). Alege o permutare aleatoare $\tau : \{1, ..., \ell\} \to \{1, .., \ell\}$ și setează criptotextul final ca

$$C=(\text{VK},U,(H_{\tau(1)},C_{\tau(1)}),...,(H_{\tau(\ell)},C_{\tau(\ell)}),\sigma),$$

  unde $\sigma = (\text{SK},U,(H_{\tau(1)},C_{\tau(1)}),...,(H_{\tau(\ell)},C_{\tau(\ell)}))$.

- Dec(BE-MPK,sk_i, C): pentru inputul C=(VK,U,(H_1, C_1),...,(H_ℓ, C_ℓ),σ) și cheia secretă sk_i, sk_i^h, returnează $\perp$ dacă $\mathcal{V}$(VK,U,(H_1, C_1), ..., (H_ℓ, C_ℓ),σ) = 0 sau dacă U nu este in spațiul definit de π^{hint}. Altfel, calculează $H = \pi^{hint}$Invert(cp,sk_i^h,VK,U). Dacă $H \neq H_j$ pentru toți $j \in \{1, .., \ell\}$ returnează $\perp$. Altfel fie j cel mai mic index astfel încât $H = H_j$ și caluculează $M' = \pi^{pke}$.Decrypt(sk_i, C_j). Dacă M' poate fi împărțit ca $M' = M||$VK pentru M de lungime potriviă, returnează M. Altfel returnează $\perp$.

Corectitudinea acestei scheme rezultă direct din corectitudinea și slab robustețea schemelor sale componente π^{hint} și π^{pke}.

Teorema 5. Construcția de mai sus este adaptiv ANO-IND-CCA-sigură dacă:

(i) : π^{hint} este anonimă;

(ii) π^{pke} este AI-CCA sigură și slab robustă la atacuri de tip criptotext ales;

(iii) Σ este semnatură one-time puternică;

6 Alternativă la criptarea cu chei publice

Criptarea de tip broadcast și criptarea cu chei publice diferă în mai multe moduri. Diferența principală este modul în care sunt alocate cheile. În sistemele de criptare de tip broadcast dispozitivele primesc o mulțime de chei și noi dispozitive pot fi usor adăugate în grupuri privilegiate de utilizatori. Nici unul dintre aceste dispozitive nu are nevoie să cunoască informații despre celelalte și tot ce știu despre este ca aparțin grupului privilegiat.

Pe de altă parte criptarea cu chei publice se bazează pe cunoașterea prealabilă a tuturor dispozitivelor. Emițătorii folosesc cheia publică a receptorilor pentru a cripta un mesaj iar receptorii își folosesc cheia secretă pentru al decripta. Deoarece aceste chei trebuie să fie cunoscute înainte de a se efectua schimbul de mesaje, înseamnă ca fiecare dispozitiv trebuie să știe despre celelalte deci trebuie să stocheze cheile celorlalte dispozitive cu care vrea să comunice.

Printre avantajele folosirii sistemelor de criptare de tip broadcast în locul celor cu chei publice se numară viteza și abilitatea de a se adapta la atacuri asupra sistemului. Din moment ce sistemele de criptare de tip broadcast efectuează operații operații simetrice simple iar sistemele de criptare cu chei publice operații de exponențiere criptarea de tip broadcast se efectuază de până la 1.000 de ori mai repede decât criptarea cu chei publice după cum se arată în [3].

După cum am menționat și anterior abilitatea de a revoca chei din sistem este un avantaj major ce conferă longevitate și durabilitate sistemului. Fără această abilitate sistemul se transformă într-un sistem de partajare a secretelor ce este spart când o cheie este descoperită de o entitate neautorizată. De exemplu sistemul de criptare și codificare al conținutului video (CSS) pentru DVD-uri nu poate fi reparat fără reproiectarea sistemului de la zero. Dacă sar fi folosit o schemă de criptare de tip broadcast creatorii ar fi putut să creeze DVD-uri noi care nu ar fi stocarea de programe piratate neavând nici un efect asupra utilizatorilor legitimi.

Protejarea drepturilor de autor a devenit o aplicație importantă pentru criptografie. În locul unui sistem în care doar utilizatorii autorizați au acces la chei, sistemele pentru drepturi de autor furnizează chei pentru toți utili-

zatorii deoarece nu se poate spune diferența. Din acest motiv sunt sensibile la inginerie inversă. Sistemele de criptare cu chei publice neceseită schimburi prealabile de chei, din acest motiv cheile sunt plasate la nivelul de legătură al codului unde pot fi foarte usor găsite de utilizatorii neautorizați. Pe de altă parte din moment ce sistemele de criptare de tip broadcast sunt într-un singur sens pot ascunde cheile mult mai adânc în program, gasirea acestora de utilizatori neautorizați find mult mai dificilă.

Criptarea de tip broadcast nu este aplicabilă pentru toate aplicațiile însă este foarte utilă în protejarea conținutului. Poate fi utilă în televiziunea platită, distribuirea informatiilor cu drepturi de autor prin CD-uri și DVD-uri, distribuirea de conținut video și audio pe internet. Costurile fixe, abilitatea de revocare și rezistența la inginerie inversă sunt foarte importante pentru electronicele de consum.

Scopul inițial al criptării de tip broadcast sa dovedit mai puțin important decât o altă aplicație a acesteia: protejarea conținutului media. Protejarea conținutului media este foarte important deoarece utilizatorii au acum acces la toate tipurile de contținut media în formă digitală. Din moment ce milioanele de copii digitale sunt la fel de importante ca originalul protejarea conținutului digital și a drepturilor creatorilor sunt o preocupare în creștere. Traw propune două metode pentru protecția conținutului: acordoarea de licențe și tehnologia [18]. Acordarea de licențe este o abordare mai eficientă pentru unii utilizatori deoarece având timp suficient un tehnician experimentat va găsi o breșă tehnologică. Cu toate acestea furnizarea sigură și eficientă de protecție bazată pe tehnologie va preveni masele de utilizatori obișnuiți sa încalce proprietatea intelectuală a altora.

Aceasta poate fi văzută ca strategie de apărare în profunzime pentru editorii importanți. Apărarea în profunzime a fost mult timp cunoscută ca o calitate benefică în domeniul securității care prevede mai multe metode de a asigura securitatea datelor în cazul în care oricare din ele se dovedește a fi deficitară. Aceste două metode de protecție a conținutului, acordarea de licențe și folosirea tehnologiei, sunt utile în combaterea utilizatorilor să acceseze date pentru care nu au drepturi. Ripley și Michael arată în [2] că deoarece calitatea unor utilizatori a crescut de la „simpli copiatori" la „pirați profesioniști" acordarea de licențe devine o abordare tot mai importantă.

7 Aplicație de streaming

În continuarea folosind sistemul ANOBE bazat pe PKE prezentat la în capitolul anterior în care pentru criptarea mesajului se foloseste cheia publică a fiecarui utilizator și o semanătură digitală, vom realiza o aplicație de streaming ce poate fi folosită pentru, streamingul de jocuri video, tutoriale live etc în care broadcasteerul poate decide care dintre categoriile de utilizatori poate avea acces la continut.

Pentru realizarea acestei aplicații sa folosit frameworkul .NET deoarece oferă suport nativ pentru funcționalitățile de bază ale acestei aplicații: crearea arhitecturii Client-Server, crearea interfeței grafice și lucrul cu interfețele grafice pentru realizarea protocolului.

Pentru implementarea protocolului sau folosit sistemul de criptare cu chei publice RSA și schema de semnare RSA puse la dispozitie de libraria System.Security.Cryptography. Pentru realizarea interfețelor grafice sa folosit Windows Forms iar pentru realizarea comunicării protocolul TCP.

Aplicatia oferă urmatoarele functionalită: autentificarea utilizatorului și a clientilor, transmisia live a streamului. După autentificarea în aplicație utilizatorul are posibiliatea de a alege căror grupuri de clienți (subscribers, fallowers...) utilizatori este dedicat streamul după care porneste streamul iar clienții care sunt în aceste grupuri îl pot urmari live.

8 Concluzii

Această lucrare ne arată faptul că există alternative în protecția confidențialității datelor în contextul transmiterii informatiei în sistemele de distribuit conținut.

Mai întâi am prezentat schema generică de criptare de tip broadcast și cateva variatii ale acesteia: sisteme de criptare bazate pe identitate, sisteme de criptare bazate pe atribute și sisteme cu anonimat. După care am trecut la a face o comparație între sistemele de criptare cu chei publice și sistemele de criptare de tip broadcast.

Sistemele de criptare de tip broadcast nu pot fi folosite în toate situațiile. Totuși sunt unele situații unde se dovedesc a fi mai bune decât sistemele de criptare cu chei publice. Un astfel de exemplu este cazul criptarii DVD-urilor. Sistemul de criptare și codificare al DVD-urilor (CSS) a fost deja spart și e foarte usor să gasim programe care care pot decripta DVD-uri criptate. Din moment ce CSS este bazat pe o schemă de partajare a secretelor nu poate fi schimbată în mod dimamic ca în cazul schemelor de criptare de tip broadcast deci breșa de securitate nu poate fi reparată. Acesta este doar unul din exemplele pentru care sistemele de criptare de tip broadcast ar trebui folosite atunci când este posibil.

Profesioniștii care implementează sisteme de securitate ar trebui să știe despre această tehnologie și să o implementeze atunci când este cazul. Sistemele de criptare de tip broadcast aduc numeroase beneficii mai ales în contextul protejării conținutului media.

Una dintre direcțiile în care sistemele de criptare de tip broadcast ar putea fi îmbunătătite este performanța. Desi sunt mai rapide decât sistemele de criptare cu chei publice de cele mai multe ori se face un compromis prin minimizarea dimensiunii criptotextului sau a spatiului cheilor. O soluție optimă ar fi cea în care atât dimensiunea criptotextului cat și spațiul mesajelor să fie minime.

Îmbunătațirea funcționalităților de securitate ale aplicațiilor trebuie luată în calcul în permanență și cercetarea în acest domeniu trebuie sprijinită pen-

tru a îmbunătăți beneficiile oferite clienților.

Bibliografie

[1] A. Fiat, M. Naor,"Broadcast encryption" , In Proceedings of Crypto '93, volume 773 of LNCS, pages 480–491. Springer-Verlag, 1993.

[2] Ripley, Michael, et. al. "Content Protection in the Digital Home." Intel Technology Journal. 15 November 2004: 48-56. 9 January 2005.

[3] Lotspiech, Jeffrey, Steffan Nusser, and Florian Pestoni. "Broadcast Encryption's Bright Future." IEEE Computer 35.8 (August 2002): 57-63.

[4] D. Boneh and A. Silverberg."Applications of multilinear forms to cryptography." Contemporary Mathematics, 324:71–90, 2003.

[5] D. Boneh, C. Gentry, and B. Waters."Collusion resistant broadcast encryption with short ciphertexts and private keys." Cryptology ePrint Archive, Report 2005/018, 2005

[6] Cécile Delerablée. "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys". *ASIACRYPT*, LNCS 4833: 200- 215, 2007

[7] Zhang L., Hu Y., and Mu N "An Identity-based Broadcast Encryption Protocol for Ad-hoc Networks," In Proceedings of The 9th International Conference for Young Computer Scientists, USA, pages 1619-1623, 2008.

[8] Liang Hu, Zheli Liu,Xiaochun Cheng "Efficient Identity-based Broadcast Encryption without Random Oracles". Journal of Computers, vol. 5, no. 3, pages. 331-336, 2010.

[9] Libert, B., Paterson, K. G., and Quaglia, E. A. "Anonymous broadcast encryption: Adaptive security and efficient constructions in the standard model." In PKC 2012 (May 2012), M. Fischlin, J. Buchmann, and M. Manulis, Eds., vol. 7293 of LNCS, Springer, pp. 206–224.

[10] Adam Barth, Dan Boneh, and Brent Waters. "Privacy in encrypted content distribution using private broadcast encryption" In G. Di Crescenzo, A. Rubin (eds.) FC 2006, volume 4107 of LNCS (Springer, Berlin, 2006), pages 52–64

[11] Cash, D., Kiltz, E., Shoup, "The Twin Diffie-Hellman Problem and Applications." In: Smart, N.P. (ed.) EUROCRYPT 2008. LNCS, vol. 4965, pages 127–145. Springer, Heidelberg (2008)

[12] Gafni, E., J. Staddon, and Y. Yin. "Efficient Methods for Integrating Broadcast Encryption and Traceability." Advances in Cryptology (Crypto 1999). Lecture Notes in Computer Science 1666. Springer-Verlag, 1999: pages 372- 387.

[13] Garay, Juan A., Jessica Staddon, and Avishai Wool. "Long-lived Broadcast Encryption." Advances in Cryptology (Crypto 2000). Lecture Notes in Computer Science 1880. Springer-Verlag, 2000: pages 333-352.

[14] B. Waters, "Eficient identity-based encryption without random oracles", Proc. of 24th Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), pages 114-127, Aarhus, Denmark, 2005.

[15] B. Malek, A. Miri, "Adaptively Secure Broadcast En yption with Short Ciphertexts", International Jour al of Network Security, vol. 14, no. 2, pages 71-79, 2012

[16] R. Canetti, S. Halevi, J. Katz, "Chosen-ciphertext security from identity-based encryption", Proc. of 23th Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'04), pages 207-222, Interlaken, Switzerland, 2004.

[17] Yevgeniy Dodis and Nelly Fazio. "Public key broadcast encryption for stateless receivers." In Proc. of Security and Privacy in Digital Right Management (DRM'02), pages 61–80, 2002

[18] Traw, Brendan S. "Protecting Digital Content Within the Home." IEEE Computer 34.10 (October 2001):pages 42-47.

[19] BSA |The Software Alliance Global Software Survey 2016 http://globalstudy.bsa.org/2016