UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ

Schema KP-ABE pentru circuite Booleene generale cu forme biliniare

propusă de

Diana Bolocan

Sesiunea: *iulie, 2019*

Coordonator științific

Prof. Dr. Țiplea Ferucio Laurențiu

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI

FACULTATEA DE INFORMATICĂ

Schema KP-ABE pentru circuite Booleene generale cu forme biliniare

Diana Bolocan

Sesiunea: *iulie, 2019*

Coordonator științific

Prof. Dr. Țiplea Ferucio Laurențiu

Avizat,

Prof. Dr. Țiplea Ferucio Laurențiu

Data: …….. Semnătura: ……

Declarație privind originalitatea conţinutului lucrării de licenţă

Subsemntatul(a) **Bolocan Diana** domiciliul în **România, jud. Galați, mun. Galați, str. Saturn, nr. 28, bl. B4, ap. 3** născut(ă) la data de **10 septembrie 1997**, identificat prin CNP **2970910170010**, absolvent(a) al(a) Universităţii „Alexandru Ioan Cuza" din Iaşi, Facultatea de **Informatică** specializarea **Informatică**, promoţia **2019**, declar pe propria răspundere, cunoscând consecinţele falsului în declaraţii în sensul art. 326 din Noul Cod Penal şi dispoziţiile Legii Educaţiei Naţionale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licenţă cu titlul: **Schema KP-ABE pentru circuite Booleene generale cu forme biliniare** elaborată sub îndrumarea **Prof. Dr. Țiplea Ferucio Laurențiu**, pe care urmează să o susţin în faţa comisiei este originală, îmi aparţine şi îmi asum conţinutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licenţă/dizertație să fie verificată prin orice modalitate legală pentru confirmarea originalităţii, consimţind inclusiv la introducerea conţinutului său într-o bază de date în acest scop.

Am luat la cunoştinţă despre faptul că este interzisă comercializarea de lucrări ştiinţifice in vederea facilitării fasificării de către cumpărător a calităţii de autor al unei lucrări de licenţă, de diploma sau de disertaţie şi în acest sens, declar pe proprie răspundere că lucrarea de faţă nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: …… Semnătura: ……...

Declarație de consimțământ

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Schema KP-ABE pentru circuite Booleene generale cu forme biliniare*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași, *24 Iunie 2019*

Absolvent *Diana Bolocan*

_____

Cuprins

Motivație

Securitatea informației a fost întotdeauna un subiect foarte interesant pentru mine datorită caracterului său evolutiv. Este un domeniu al informaticii necesar și răspândit pretutindeni. Importanța acestuia îl face indispensabil oricărei persoane, iar modalitățile de abordare a problemelor implică mereu creativitate și perspective noi de rezolvare.

Din punctul meu de vedere, printre cele mai fascinante subiecte din securitatea informației se numără criptografia, deoarece reușește să îmbine limitările fizice ale mașinii de calcul cu artificii matematice. O altă caracteristică din criptografie care m-a atras a fost faptul că se încearcă găsirea unor modele cât mai sigure și eficiente. Inițial, s-ar crede că un sistem criptografic trebuie să fie perfect, impenetrabil teoretic, însă, deși idealistic se încearcă acest lucru, un model cu dezavantaje neglijabile e suficient pentru a asigura practic securitatea necesară. Această necesitate pentru aplicabilitate reprezintă una dintre principalele motive pentru care mi-am dorit să lucrez în acest domeniu.

Astfel doresc să abordez un subiect foarte interesant din securitatea informației și anume criptarea după atribute cu politici de chei, cunoscut și după acronimul KP-ABE[1], în contextul circuitelor Booleene. Scopul lucrării de licență este de a găsi un model de securitate care poate fi aplicat pe circuite Booleene generale.

După cum se poate deduce din denumirea schemei, KP-ABE este un model de securitate bazat pe criptare și decriptare după caracteristici. Un astfel de model se folosește cu scopul de a evita partajarea cheii de decriptare de la un utilizator la altul, creându-se astfel grupuri de utilizatori cu aceleași atribute care au acces individual la funcționalitatea modelului.

O soluție pentru generalizarea schemei criptografice îmi pare foarte importantă deoarece un circuit Boolean bazat pe o funcție Booleană nu poate reprezenta orice structură de acces. Astfel mi-am propus să găsesc o abordare care să fie atât viabilă cât și eficientă.

Două dintre modelele care generalizează schema criptografică KP-ABE pentru circuite Booleene sunt discutate pe parcursul lucrării de licență. Una dintre ele, și anume lucrarea [1], este în același timp articolul de bază de la care pornește lucrarea de licență.

[1] Key Policy Attribute Based Encryption

Ambele articole au ajutat la construcția noului model criptografic ce va fi prezentat în Capitolul 3.

Introducere

Problema securității informației a apărut încă din primele zile ale comunicării, dându-l ca exemplu pe Iulius Cezar care folosește cifrul cu deplasare pentru a-și păstra mesajele confidențiale încă din anul 50 î.Hr.. Această necesitate pentru secretizarea informației a devenit din ce în ce mai importantă în mod special în perioadele de război. Printre mariile realizări în domeniul securității informației putem aminti Mașina Enigma, mecanism electromagnetic criptografic cu rotoare capabil de a cripta și decripta mesaje secrete.

Actualmente, mașinile de calcul au devenit obiecte indinspesabile oricărei persoane și prin urmare securitatea informației se află pe același nivel de necesitate. Dezvoltarea tehnologică rapidă este în acord cu necesitățle și așteptările contextului internațional. Cererea imensă a pieței, concurența dintre companiile rivale au dus la apariția unei puteri computaționale mari și de calitate la prețuri accesibile. Mașinile de calcul sunt mai la îndemână ca niciodată, fie prin achiziție directă din magazine, fie prin serviciile Cloud oferite și cu toate acestea, schimbările continue și rapide fac ca acestea să rămână susceptibile la atacuri de securitate.

Toate asptectele menționate mai sus duc la concluzia că securitatea informației va rămâne un domeniu solicitat și de o importanță mare, cu o gamă variată și constantă de probleme ce trebuie rezolvate. Astfel, îmi încep lucrarea de licență cu scopul de a generaliza și eficientiza pe cât posibil un aspect din acest larg domeniu menționat, și anume schemele criptografice KP-ABE în circuite Booleene prezentate în articolul de specialitate [1]. Noua abordare propusă se bazează atât pe aplicare de principii fundamentale din criptografie, cât și pe modificarea structurii fizice a circuitelor.

În Capitolul 1 se vor detalia noțiunile folosite în noua abordare, menite a fi suport teoretic pentru deciziile luate în privința modalității de soluționare a problemei generalizării și eficentizării sistemului criptografic.

În Capitolul 2 se vor prezenta articolele care stau la baza lucrării de licență, abordându-se subiecte precum complexitatea sistemului criptografic, soluționarea atacului Backtracking (atac foarte important care îngreunează generalizarea schemelor

criptografice în circuitele Booleene), obiectivele propuse a fi atinse și alte abordări întâlnite.

Capitolul 3 conține soluția propusă, alături de explicații pentru deciziile făcute, demonstrații de securitate, complexitatea noii scheme și o concluzie în care se vor evidenția diferențele dintre abordările menționate mai sus.

Capitolul 1

Concepte teoretice

Înainte de a începe descrierea lucrării de licență se vor prezenta noțiuni teoretice care stau la baza lucrării și care vor ajuta la buna înțelegere a acesteia. În acest capitol se vor prezenta noțiuni teoretice din domeniul criptografic și probabilistic și informații despre circuite Booleene.

1.1 Criptografie

Grup ciclic multiplicativ. Un grup este o tuplă $(G,\cdot)$, unde G este o mulțime și $\cdot$ este o operație binară, care satisface următoarele proprietăți:

$$\forall x, y \in G, x \cdot y \in G$$

$$\exists e \in G \; a.\hat{\imath}. \forall x \in G, e \cdot x = x \cdot e = x$$

$$\forall x \in G, \exists y \in G \; a.\hat{\imath}. \; x \cdot y = y \cdot x = e$$

$$\forall x, y, z \in G, (x \cdot y) \cdot z = x \cdot (y \cdot z)$$

Un grup se numește **multiplicativ** dacă operatorul $\cdot$ este operatorul de înmulțire.

Un grup este **ciclic** dacă prin aplicări repetate a operatorului $\cdot$ se pot obține toate elementele grupului. Un astfel de element particular din care se poate construi tot grupul se numește **generator**, notat g.

Forme biliniare. Fie G_1, G_2, G_3 trei grupuri ciclice de același ordin. O formă biliniară de la $G_1 \times G_2$ la G_3 este o funcție $e: G_1 \times G_2 \to G_3$ unde pentru $\forall u \in G_1, v \in G_2, \; a, b \in \mathbb{Z}$,

$$e(u^a, v^b) = e(u, v)^{ab}.$$[2]

Problema logaritmului discret. Logaritmii discreți sunt logaritmi definiți în contextul grupurilor ciclice multiplicative. Fie g un generator al grupului ciclic

[2] https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/bilinear-maps.pdf

multiplicativ G. Știm că fiecare element al lui G poate fi scris ca g^{α}. Problema logaritmului discret este definită astfel: dat g, G și $x, x \in G$ să se găsească α (unde $\alpha = \log_g x$).[3] Există grupuri pentru care rezolvarea problemei este dificilă (precum $\mathbb{Z}_p^*$, unde p este un număr prim mare). Grupuri preferate folosite în modelele criptografice sunt $\mathbb{Z}_p^*$ și subgrupurile ciclice ale curbelor eliptice peste corpuri finite.

Problema decizională biliniară Diffie-Hellman. **Problema computațională Diffie-Hellman** spune că, date fiind g^a și g^b cu a, b alese uniform aleator și independente între ele din $\mathbb{Z}_p$, valoarea g^{ab} arată ca un element oarecare din G, grup multiplicativ ciclic. **Problema decizională Diffie-Hellman** are pe lângă valorile de mai sus și pe g^c, cu c ales de asemenea aleator uniform și independent din $\mathbb{Z}_p$ și arată că (g^a, g^b, g^{ab}) și (g^a, g^b, g^c) sunt computațional indisitingibile. **Problema decizională biliniară Diffie-Hellman** în grupul bilinear G_2 este problema distingerii dintre $e(g,g)^{abc}$ și $e(g,g)^z$ date fiind g, g^a, g^b, g^c cu g generator pentru G_1, a, b, c, z valori aleator alese din $\mathbb{Z}_p$ și forma biliniară $e: G_1 \times G_1 \to G_2$. **Ipoteza problemei decizionale biliniară Diffie-Hellman** în G_2 spune că nu există algoritm de timp polinomial care să rezolve problema în G_2 cu un avantaj neglijabil [1].

Structuri de acces. Fie U o mulțime de elemente care se numesc în acest caz **atribute**. O mulțime S de submulțimi nenule ale lui U se numește **structură de acces autorizată** dacă are ca scop definirea utilizatorilor autorizați într-un sistem criptografic. Orice altă submulțime care nu aparține lui S reprezintă un utilizator **neutorizat**. Structura de acces neautorizată se notează de obicei cu $\bar{S}$.

Adversar. Din punct de vedere criptografic, un adversar este un algoritm cu putere computațională nelimitată sau un algoritm cu putere computațională limitată, precum un algoritm de timp polinomial.[4] Adversarul este văzut ca o entitate malițioasă care are ca scop principal trecerea de o proprietate de securitate pentru a descoperi un secret sau corupe de date ș.a.m.d.

Oracol. Un oracol este o entitate a cărei securitate se încearcă a se demostra cu ajutorul unui adversar. Jocul criptografic dintre oracol și adversar se bazează pe schimb

[3] https://www.doc.ic.ac.uk/~mrh/330tutor/ch06s02.html
[4] Prof. Dr. Ferucio Laurețiu Țiplea. Curs Introducere în Criptografie. Perfect Security

de răspunsuri, unde adversarul încearcă să găsească informații utile din interacțiunea cu oracolul.

Automat finit. Un automat finit este un sistem format din noduri numite stări și arce către noduri care indică tranziția de la o stare la alta ca răspuns la intrări externe.

Politică de securitate. O politică de securitate poate fi reprezentată ca o partiție a operațiilor în operații autorizate și operații neautorizate, operații ce schimbă starea unui automat finit.

Interpolare polinomială. Interpolarea este o metodă de generare prin aproximație a unor noi valori în funcție de un set de date predefinite. Interpolarea polinomială aproximează o funcție polinomială de grad minim

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0 x^0$$

care să treacă prin punctele din setul de valori primite $\{(x_0, y_0), \dots, (x_n, y_n)\}$

$$P(x_i) = y_i, i \in \{0,1, \dots, n\}, n \in \mathbb{N}$$

Schema Shamir. Schema Shamir este un model de partajare de secrete care are la bază interpolarea polinomială. Date oricare k perechi de forma $(x_1, y_1), \dots, (x_k, y_k)$, cu $x_i \neq x_j, 1 \leq i < j \leq k$, există o singură funcție polinomială $P(x)$ de grad $k-1$ astfel încât $P(x_i) = y_i, 1 \leq i \leq k$. Partajările $I_1, \dots, I_k$ sunt alese ca $I_i = P(x_i), 1 \leq i \leq n$, unde $x_1, \dots, x_n$ sunt distincte două câte două.

Pentru partajările $\{I_i \mid i \in A\}$, unde A este o mulțime de k elemente, secretul poate fi reconstruit folosindu-se formula lui Lagrange de interpolare

$$S = \sum_{i \in A} (I_i \cdot \prod_{j \in A/\{i\}} \frac{x_i}{x_j - x_i}).$$ [5]

Secvențe Mignotte. Fie n un număr natural mai mare sau egal cu 2, și $2 \leq k \leq n$. O secvență (k, n) Mignotte este o secvență de numere întregi coprime două câte două $p_1 < p_2 < \cdots < p_n$ care satisfac proprietatea

$$\prod_{i=0}^{k-2} p_{n-i} < \prod_{i=1}^{k} p_i.$$

[5] Sorin Iftene. Secret Sharing Schemes with Applications in Security Protocols. Ph.D. Thesis. Din ianuarie 2007

Teorema chineză a resturilor. Fie $k \geq 1$ și $m_1, m_2, \dots, m_k$ numere întregi coprime două câte două. Pentru oricare $b_1, b_2, \dots, b_k \in \mathbb{Z}$ sistemul S de ecuații are o soluție unică modulo $m_1 \cdot m_2 \cdot \dots m_k$:

$$S: \begin{cases} x \equiv b_1 \ mod \ m_1 \\ \dots \\ x \equiv b_k \ mod \ m_k \end{cases}$$

Soluția poate fi calculată astfel:

1. se asignează pentru fiecare c_i cu $1 \leq i \leq k$:

$$c_i = \prod_{j=1, j \neq i}^{k} m_j$$

2. se calculează o soluție întreagă x_i pentru ecuația $c_i x \equiv b_i \ mod \ m_i, \forall i$
3. soluția unică modulo $m_1 \cdot m_2 \cdot \dots m_k$ a sistemului este:

$$x \equiv (c_1 x_1 + \dots + c_k x_k) \ mod \ (m_1 \cdot m_2 \cdot \dots m_k).$$ [6]

Schema Mignotte. Schema Mignotte este un model de partajare de secrete care are la bază secvențe speciale de numere întregi numite **secvențe Mignotte**. Dată o secvență Mignotte, schema funționează astfel:

1. se alege ca secret un număr întreg aleator astfel încât $\beta < S < \alpha$, unde $\beta = \prod_{i=0}^{k-2} p_{n-i}$ și $\alpha = \prod_{i=1}^{k} p_i$
2. secretele partajate sunt calculate după $I_i = S \ mod \ p_i, 1 \leq i \leq n$
3. date k secrete partajate $I_{i_i}, \dots, I_{i_k}$, secretul S poate fi reconstruit folosindu-se teorema chineză a resturilor pentru sistemul modulo $p_1 \cdot p_2 \cdot \dots \cdot p_k$ cu soluție unică

$$\begin{cases} x = I_{i_1} \ mod \ p_{i_1} \\ \dots \\ x = I_{i_k} \ mod \ p_{i_k} \end{cases}.$$ [7]

[6] Prof. Dr. Ferucio Laurețiu Țiplea. Curs Fundamente Algebrice ale Informaticii. Computational Introduction to Number Theory Part II

[7] Lect. Dr. Sorin Iftene. Secret Sharing Schemes with Applications in Security Protocols. Ph.D. Thesis. Din ianuarie 2007

1.2 Circuite Booleene

Circuit Boolean. Un circuit Boolean este un model matematic pentru circuite logice digitale compus din porți logice și fire. Porțile întâlnite într-un circuit sunt: $AND, OR, NOT, NAND, NOR, XOR$ și $XNOR$. Un circuit Boolean se numește **monoton** dacă este compus numai din porți logice AND și OR. În funcție de componența circuitul Boolean, acesta poate fi rescris folosindu-se **legea De Morgan**.

Legea De Morgan. În logica propozițională și algebra Booleană, legile lui De Morgan sunt o pereche de reguli de transformare care permit conjuncțiilor și disjuncțiilor să fie rescrise folosind negația:

$$\overline{A \cap B} = \bar{A} \cup \bar{B}$$

$$\overline{A \cup B} = \bar{A} \cap \bar{B}.$$

Astfel o poartă logică AND poate fi rescrisă ca NOT aplicat pe OR (sau NOR) și o poartă logică OR poate fi rescrisă ca NOT aplicat pe AND (sau $NAND$). Această lege va fi folosită mai târziu în descrierea soluției lucrării de licență.

Fanout. Fanout-ul este definit ca numărul maxim de fire care por fi conectate la ieșirea unei porți logice.

1.3. Probabilități

Experiment aleator. Se numește **experiment aleator** un experiment al cărui rezultat nu este cunoscut dinainte, dar ale cărui rezultate posibile sunt cunoscute în totalitate și care poate fi repetat în condiții identice.

Spațiu de selecție. Un rezultat posibil al unui experiment aleator se numește **eveniment aleator elementar**, iar mulțimea acestora se numește **spațiu de selecție** sau **al evenimentelor elementare** notat cu Ω.

Eveniment aleator. Se numește **eveniment aleator** o anumită submulțime a spațiului de selecție: $A \subseteq \Omega$.[8]

[8] Lect. Dr. Olariu Florentin Emanuel. Curs Probabilități și Statistică. Cursul 1

Variabilă aleatoare. Dat un experiment aleator ε și Ω mulțimea evenimentelor aleatoare elementare, o **variabilă aleatoare** reală este o funcție $X: \Omega \to \mathbb{R}$, astfel încât pentru orice interval $J \subseteq \mathbb{R}$, $X^{-1}(J)$ este un eveniment aleator.[9]

Distribuție aleatoare uniformă. O variabilă aleatoare se spune că este distribuită **uniform cu parametrul $n \in \mathbb{N}$** dacă are repartiția:

$$U_n: \begin{pmatrix} 1 & 2 & \dots & n \\ 1/n & 1/n & \dots & 1/n \end{pmatrix}.$$ [10]

Probabilități condiționate. Fie A și B două evenimente aleatoare, **probabilitatea condiționată** de a se realiza A știind că s-a realizat B este

$$P(A|B) = \frac{P(A \cap B)}{P(B)}.$$

Formula probabilității totale. Fie $\{A_1, A_2, \dots, A_n\}$ evenimente aleatoare care realizează o partiție a evenimentului sigur Ω

$$\bigcup_{i=1}^{n} A_i = \Omega$$

$$A_i \cap A_j = \emptyset, \forall i \neq j.$$

Dacă B este un eveniment oarecare, atunci

$$P(B) = \sum_{i=1}^{n} P(B|A_i) \cdot P(A_i),$$

dacă toate evenimentele care condiționează sunt posibile. [11]

[9] Lect. Dr. Olariu Florentin Emanuel. Curs Probabilități și Statistică. Cursul 3
[10] Lect. Dr. Olariu Florentin Emanuel. Curs Probabilități și Statistică. Cursul 4
[11] Lect. Dr. Olariu Florentin Emanuel. Curs Probabilități și Statistică. Cursul 2

Capitolul 2.

Sisteme criptografice KP-ABE, extensie pentru circuite Booleene

Criptarea bazată pe atribute de acces (ABE) este un tip de criptare pe chei publice, în care identitatea utilizatorului este definită ca un set de atribute (exemplu: rolurile unui angajat într-o companie), fiind subclasată în key-policy ABE (KP-ABE) și în chiphertext-policy ABE (CP-ABE). [12] După cum se poate deduce din nume, această schemă criptografică se folosește de caracteristici pentru a cripta și decripta texte. Scopul acestora este de a creea grupuri de utilizatori care au acces individual la funcționalitatea modelului fără a mai fi nevoie de partajarea cheilor de decriptare între aceștia.

În CP-ABE [2], informația criptată poate fi accesată numai de utilizatorii ale căror credențiale satisfac politica de securitate. O politică de securitate poate fi definită peste atribute folosind conjuncții, disjuncții și porți (k, n), unde k reprezintă numărul de atribute care trebuie să fie prezente dintre cele n. Autorizația este inclusă în mesajul încriptat, astfel permisiunea realizându-se implicit.

În KP-ABE [3], politica de access este codificată în cheia secretă a utilizatorului, mesajul criptându-se după setul de atribute. Fiecărei chei îi este asociată o structură de acces care specifică ce tip de criptotext poate decripta. Structura de acces folosită este un arbore ale cărui noduri frunze sunt toate elementele din setul de atribute. Fiecare criptotext are etichetat un set de atribute descriptive.

2.1 Descrierea extensiei

În lucrarea de bază [1] a licenței se demonstrează posibilitatea acomodării schemei KP-ABE pentru circuite Booleene (monotone) folosindu-se forme biliniare. Sistemul criptografic are la bază patru algoritmi pentru inițializare, criptare, generare de chei și decriptare:

[12] https://crypto.stackexchange.com/questions/17893/what-is-attribute-based-encryption

- $Setup(\lambda, n)$ este primul algoritm apelat în sistem, care primește ca parametru de intrare λ, un parametru de securitate, ce va returna un set de parametrii publici (PP) și o cheie secretă (MSK). Se folosește parametru λ pentru alegerea unui număr prim p, a două grupuri multiplicative G_1 și G_2 de ordin p, a unui generator g din G_1 și a unei forme biliniare $e: G_1 \times G_1 \to G_2$. Se definește setul $U = \{1, 2, \dots, n\}$ de atribute, $y \in \mathbb{Z}_p$ și $t_i \in \mathbb{Z}_p$ pentru fiecare atribut i din U. La final, setul de parametrii arată în felul următor:

$$PP = \left(p, G_1, G_2, g, e, n, Y = e(g,g)^y, (T_i = g^{t_i} | i \in U)\right)$$

$$MSK = (y, t_1, \dots, t_n)$$

- $Encrypt(m, A, PP)$ este algoritmul de criptare a mesajului primit m peste setul de atribute $A \subseteq U$ în care se alege un $s \in \mathbb{Z}_p$ și se returnează

$$E = (A, E' = mY^s, (E_i = T_i = g^{t_i s}), g^s)$$

- $KeyGen(C, MSK)$ generează o cheie de decriptare D pentru structura de acces definită de circuitul C pentru n inputuri astfel:
  1. Se generează (S, P) din $Share(y, C)$, unde $Share(y, C)$ reprezintă un algoritm ce returnează o listă de partajări $S(i)$ pentru firele de intrare i asociate secretului y, ca în Fig. 1.
  2. Se returnează $D = ((D(i) | i \in U), P)$, unde

     $D(i) = \left(g^{\frac{S(i,j)}{t_i}} \middle| 1 \le j \le |S(i)| \right)$, pentru fiecare $i \in U$.

- $Decrypt(E, D)$ primește ca parametrii valorile E și D definite anterior și decriptează astfel:
  1. Se calculează $V_A = (V_A(i) | i \in U)$, unde $V_A(i, j) = e\big(E_i, D(i, j)\big) = e\left(g^{t_i s}, g^{\frac{S(i,j)}{t_i}}\right) = e(g,g)^{S(i,j)s}$, pentru fiecare $i \in A$ și $1 \le j \le |S(i)|$, și $V_A(i)$ este o listă de $|S(i)|$ simboluri $\perp$, pentru fiecare $i \in U - A$.
  2. Se generează R o listă de valori în $G_2 \cup \{\perp\}$ asociate fiecărui fir de intrare prin $Recon(C, P, V_A, g^s)$, algoritm determinist care returnează rezultatul evaluării în setul de atribute V_A.
  3. Se calculează $m = E'/R(o, 1)$, unde o este firul de ieșire al circuitului C.

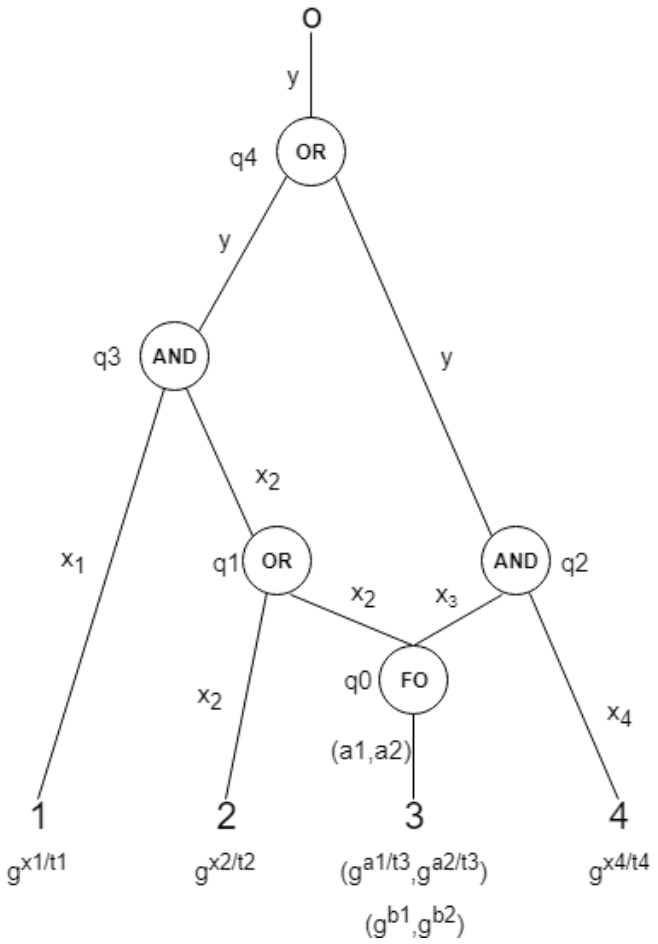

Fig. 1. Algortimul $Share$ în circuitul Boolean [1]

Procesul de construcție $Share(y, C)$ pe porțile circuitului Boolean se realizează începând de la firul de ieșire ce conține ca valoare pe y și continuând în jos spre firele de intrare astfel:

- Pentru (w_1, w_2, OR, w) și $S(w) = L$:

$$S(w_1) = L$$
$$S(w_2) = L$$

- Pentru (w_1, w_2, AND, w) și $S(w) = L$: pentru fiecare element $l \in L$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează x_l^2 astfel încât $l = (x_l^1 + x_l^2) \; mod \; p$.

$$S(w_1) = (x_l^1 | l \in L)$$
$$S(w_2) = (x_l^2 | l \in L)$$

- Pentru $\left(w, FO, w_1, \dots, w_j\right)$ și $S(w_k) = L_k$, $1 \leq k \leq j$: pentru fiecare $l_k \in L_k, 1 \leq k \leq j$, se alege aleator uniform $x_{l_k}^1 \in \mathbb{Z}_p$ și se calculeaza $x_{l_k}^2$ astfel încât $l_k = \left(x_{l_k}^1 + x_{l_k}^2\right) mod \; p$.

$$L'_k = \left(x_{l_k}^1 \middle| l_k \in L_k, 1 \le k \le j\right)$$

$$P(w_k) = \left(g^{x_{l_k}^2} \middle| l_k \in L_k\right)$$

$$S(w) = L'_1 \dots L'_j$$

Pentru o mai bună înțelegere a algoritmului $Share$, acesta a fost aplicat pe exemplul de circuit Boolean prezentat în Fig. 1. După cum se poate observa, porțile OR trimit mai departe valorile primite pe ambele ramuri, iar cele AND și FO creează noi valori pentru fiecare element din lista primită.

Procesul de reconstrucție $Recon(C, P, V, g^s)$ se realizează începând de la firele de intrare și construind valorile spre firul de ieșire astfel:

- $R(i) = V(i)$, pentru fiecare $i \in U$
- Pentru (w_1, w_2, OR, w) cu $R(w_1)$ și $R(w_2)$ definite se asignează
$$R(w, i) = \sup\{R(w_1, i), R(w_2, i)\}$$
- Pentru (w_1, w_2, AND, w) cu $R(w_1)$ și $R(w_2)$ definite se asignează
$$R(w, i) = R(w_1, i) \cdot R(w_2, i)$$
- Pentru $(w, FO, w_1, \dots, w_j)$ cu $R(w)$ definit se fac următorii pași:
  - Se împarte $R(w)$ în j liste $R(w) = R_1 R_2 \dots R_j$ cu $|R_k| = |P(w_k)|$ pentru fiecare $1 \le k \le j$
  - Se asginează $R(w_k, i) = R_k(i) \cdot e(P(w_k, i), g^s)$ pentru fiecare $1 \le k \le j$ și $1 \le i \le |R_k|$

Soluția propune adăugarea unor porți ajutătoare fanout (FO), care elimină șansa atacului backtracking, detaliat mai pe larg în următoarea secțiune. Prevenția acestuia se realizează prin funcționalitatea porții care are la bază partajare de secrete. După cum se poate observa din descrierea acesteia, poarta FO creează pentru fiecare valoare nouă încă două elemente noi.

2.2 Atacul Backtracking

Atacul backtracking reprezintă o exploatare a scurgerii de informații din poarta OR fiind discutat în lucrările [1] și [5], care apare în circuitele Booleene în care poarta este conectată la un fanout mai mare de unu.

Prezentarea atacului backtracking se va face cu ajutorului circuitului Boolean din Fig. 2. de mai jos.

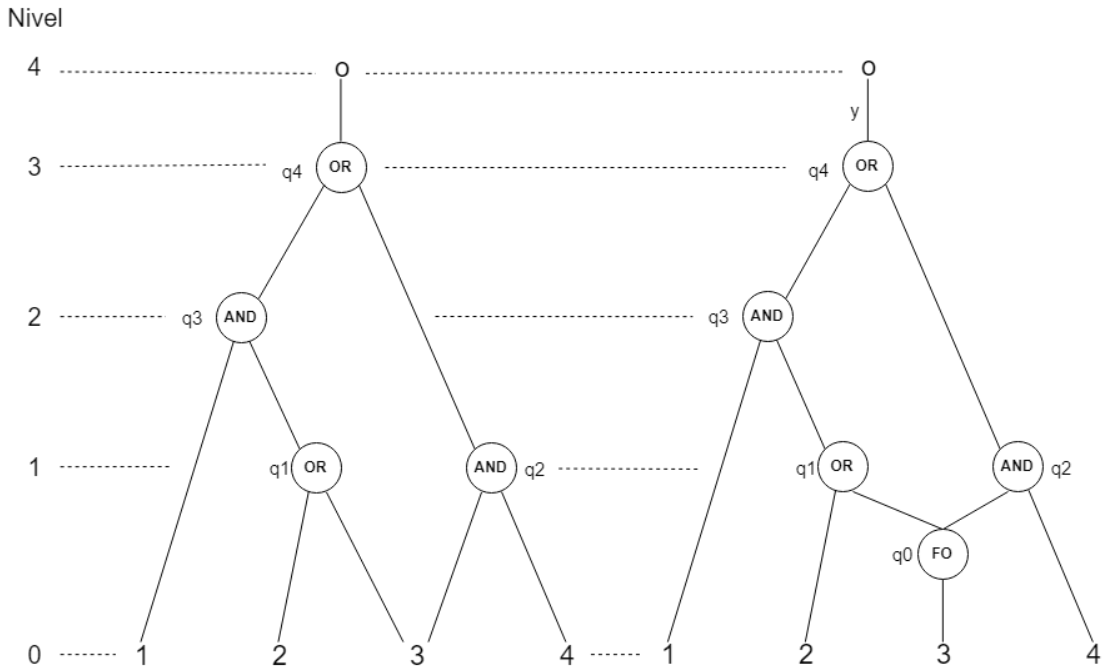

Fig. 2. a). Circuit Boolean în care apare atacul backtracking [1]

Fig. 2. b). Circuit Boolean în care atacul backtracking este soluționat prin folosirea porții ajutătoare FO [1]

În procesul de construcție $Share$, poarta OR, Γ_1, trimite mai departe valoarea din firul w către firele de intrare w_1 și w_2. Dacă se încearcă totuși o decriptare pe atributele $\{1,2,4\}$ în circuitul Boolean din Fig. 2. a)., utilizatorul este autorizat și primește cheia de decriptare, ceea ce nu ar trebui să se întâmple. Motivul pentru care acest lucru este posibil se datorează modului în care este definită poarta OR.

Pentru atributul 2 se cunoaște valoarea $e(g,g)^{sr_2}$ și știm că $r_2 = r_3$ din funcționalitatea porții OR descrisă mai sus, deci $e(g,g)^{sr_2} = e(g,g)^{sr_3}$. Astfel, valoarea calculată la firul de intrare pe atributul 2 „migrează” la atributul 3 prin poarta OR. Acest lucru încalcă scopul modelului criptografic, numai seturile de atribute $\{\{1,3,4\}, \{1,2,3,4\}\}$ fiind autorizate în acest caz.

După cum se poate vedea mai sus, atacul backtracking apare doar atunci când există un fanout mai mare de unu în circuit, altfel nu ar reprezenta o problemă de securitate. Două modalități de prevenire a atacului sunt limitarea fanout-ului circuitului

Boolean la unu, întâlnit în soluția din lucrarea [4] (discutată în secțiunea 1.4 a lucrării de licență), sau modificarea sistemului criptografic prin adăugarea unei porți FO ca în Fig. 2. b). din articolul [1].

Motivul pentru care poarta FO rezolvă problema atacului backtracking reiese din funcționalității ei. Deoarece aceasta partajează fiecare valoare ce corespunde unic unei etichete în câte două noi valori, scurgerea de informație nu mai este posibilă.

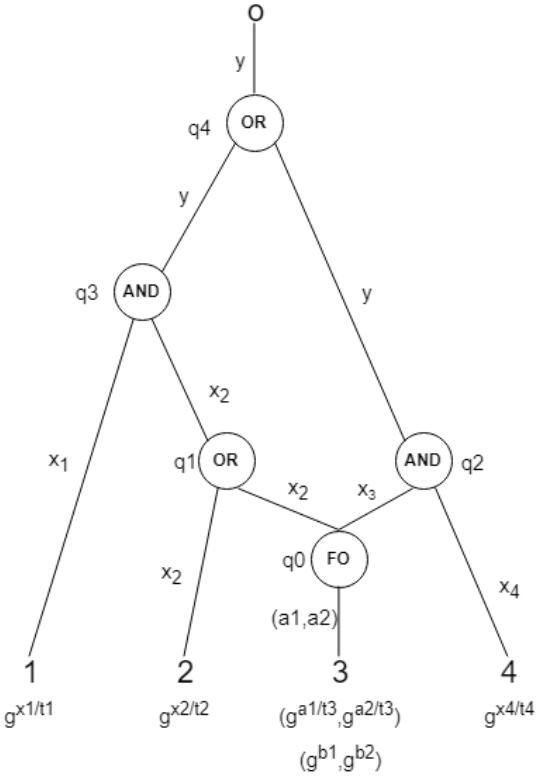

Fig. 3. a). Circuit Boolean cu poartă FO[1]

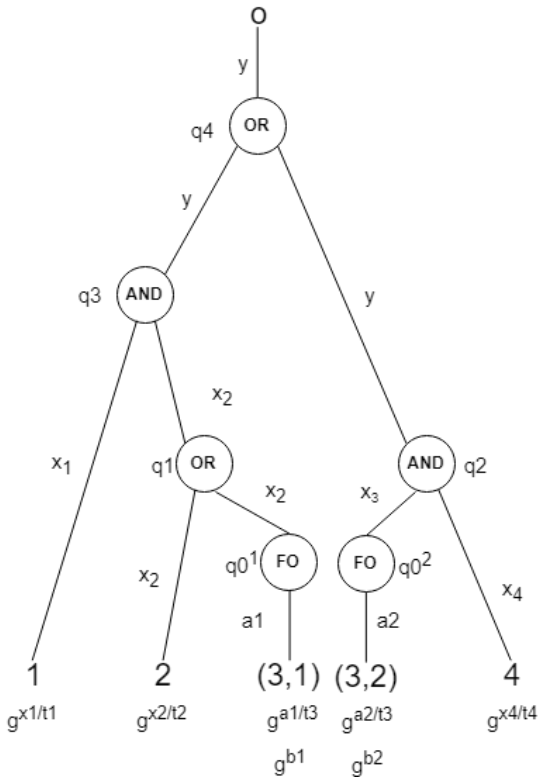

Fig. 3. b). Arborele de acces pentru circuitul alăturat[1]

Mai sus avem reprezentat arborele de acces în Fig. 3. b). pentru circuitul Boolean de la Fig. 3. a). în care se poate observa separarea valorilor pentru fanout mai mare de unu.

2.3 Complexitate

După cum se poate observa din funcționalitatea schemei criptografice, cea mai costisitoare structură o reprezintă cheia de decriptare. Mărimea acesteia depinde strict de complexitatea circuitului Boolean, crescând rapid în prezența porților FO.

Pentru a putea discuta de complexitatea schemei, presupunem că circuitul Boolean are n fire de intrare și r porți FO cu fanout de cel mult j. Schema criptografică poate fi discutată din două situații cazuale:

- Cel mai fericit caz de complexitate este atunci când nu exista porți FO pe același drum. Din cele n fire de intrare, r o să aibă cel puțin două și cel mult j valori partajate, restul având strict una, în final cheia de decriptare având $n + r(j - 1)$ componente.
- Cel mai puțin favorabil caz este atunci când există porți FO pe același drum. Poarta FO de pe nivelul cel mai înalt va trimite cel mult j partajări către o poartă FO pe un nivel mai jos, care la rândul ei va trimite cel mult j^2 partajări. Atfel, la final vor exista cel mult j^α partajări trimise către unele firele de intrare, unde α reprezintă numărul de nivele ce conțin porți FO.

Este evident că un circuit Boolean complex poate genera un număr semnificativ de elemente și că această abordare poate deveni foarte costisitoare din punct de vedere al memoriei. O soluție pentru reducerea numărului de valori este discutată mai jos în următoarea secțiune, unde abordarea implică modificări la structura circuitului Boolean.

2.4 Alte abordări

La prima vedere, o soluție foarte ușoară (și care s-a dovedit a fi mai eficientă decât schema criptografică prezentată anterior) este separarea forțată a firelor, limitându-se fanout-ul la unu. O astfel de abordare se detaliază în lucrarea [4], care are la bază articolul de specialitate [1]. Contribuția adusă în această lucrare constă în trecerea tuturor fanout-urilor mai mari decât unu în fanout-uri de strict unu (exemplu Fig. 4.), renunțându-se la porțile ajutătoare FO. Schema criptografică încearcă pe cât de mult posibil să păstreze funcționalitățile porților neschimbate.

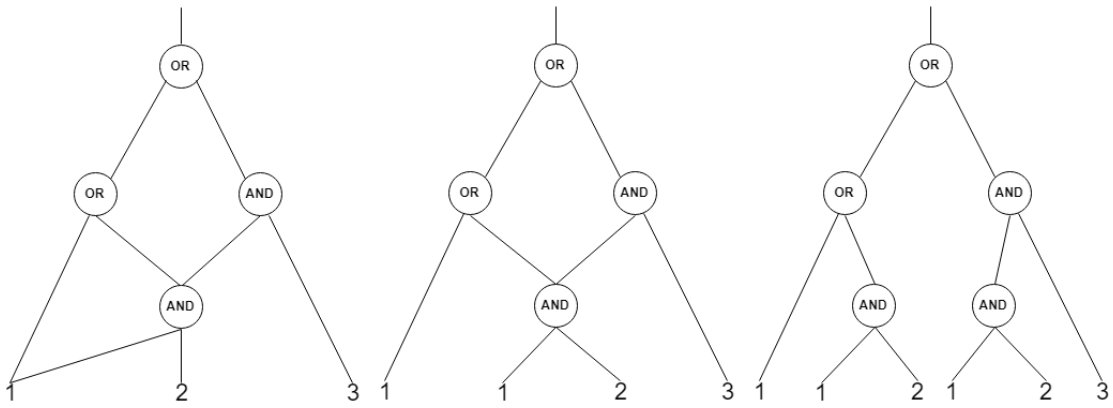

Fig. 4. Reconstrucția circuitului Boolean [4]

Expandarea circuitului Boolean se face de jos în sus, prin repetare de porți și atribute. Soluția, deși mai eficientă decât abordarea prezentată în lucrarea [1] din punct de vedere al numărului de elemente necesare generării cheii de decriptare, este costisitoare fizic datorită restructurării circuitului. În exemplul de mai sus (Fig. 4.), se poate observa că circuitul Boolean inițial conține trei atribute și patru porți cu două fanout-uri ca mai apoi să fie explicitat și să ajungă la șase atribute într-un circuit cu cinci porți.

Complexitatea cheii de decriptare în schema criptografică de mai sus este în cel mai bun caz $n + r\ (j - 1)$ și în cel mai rău caz $n + \ j^r$ și nu mai există lista de valori partajate $P(w)$.

2.5 Concluzii

Deși abordările anterioare sunt soluții valide, acestea nu sunt neapărat viabile în momentul aplicării lor peste circuite Booleene complexe. Este cât se poate de evidentă necesitatea unei soluții care să fie eficientă atât din punct de vedere al mărimii cheii de decriptare, cât și al reprezentării structurii de acces. De aceea, o soluție care reușește să combine aceste două aspecte este mai mult decât bine venită.

Capitolul 3.
Contribuție: o nouă schemă KP-ABE

După cum s-a putut observa pe parcursul prezentării lucrării, cea mai mare dificultate întâlnită în schemele criptografice descrise a fost impusă de fanout-ul în porțile OR. Această problemă însă nu apare în cazul porților AND. De aceea, mi-am propus să contribui cu o soluție care s-a dovedit a fi simplă și care reușește să facă schema KP-ABE utilizabilă în cazul circuitelor Booleene generale fără a adăuga variabile în plus ca în cazul lucrării [1] și fără a mări structura circuitului ca în cazul lucrării [4], bazându-mă pe siguranța funcționalității porților AND.

Noua schema criptografică are la bază schimbarea porților OR cu porți noi $NAND$, porți funcțional asemănătoare cu cele AND. Schimbările aduse la algoritmii noului model, precum $Share$ și $Reconstruct$, vor fi descrise la secțiunea 3.2.

3.1 Obiectiv

Porțile FO elimină șansa atacului backtracking, dar sunt foarte costisitoare pentru generarea cheii de decriptare și reprezintă operații în plus pentru întărirea sistemului criptografic. Un circuit Boolean complex ar putea duce la o creare semnificativă de valori.

Abordarea anterioară descrisă mai sus, are de asemenea dezavantajele ei. Deși este mai eficientă, aduce adăugări consistente în structura circuitului Boolean, multiplicându-se porți și atribute în încercarea menținerii securității.

Astfel, luând în seamă aceste considerente, se dorește eficientizarea pe cât de mult posibil a schemei prin reducerea/menținerea structurii fizice a circuitului și a mărimii cheii de decriptare.

3.2 Schimbarea fizică a circuitului

În toate cele trei lucrări [1], [4] și [5] se vorbește despre problema atacului backtracking în circuitele în care porțile OR au legatură cu un fanout mai mare de unu. În

articolul [1], prevenția se face prin securizarea fanout-ului cu ajutorul porților FO, în articolul [4] fanout-ul este forțat să aibă mărimea de strict unu expandând circuitul Boolean. În niciuna dintre cele trei lucrări nu se aduc modificări la poarta OR și nici nu se încearcă înlocuirea acesteia cu o alternativă mai sigură. De aceea, obiectivul acestei lucrări de licență este de a reuși evitarea atacului backtracking prin modificări aduse la porțile problematice OR. S-a putut observa că această problemă apare strict din cauza funcționalității porții în situații cazuale. Ceea ce se propune în această secțiune este schimbarea fizică a circuitului.

Modificările aduse la circuitele Booleene sunt pe cât se poate de simple. Toate porțile OR sunt înlocuite cu cele $NAND$.

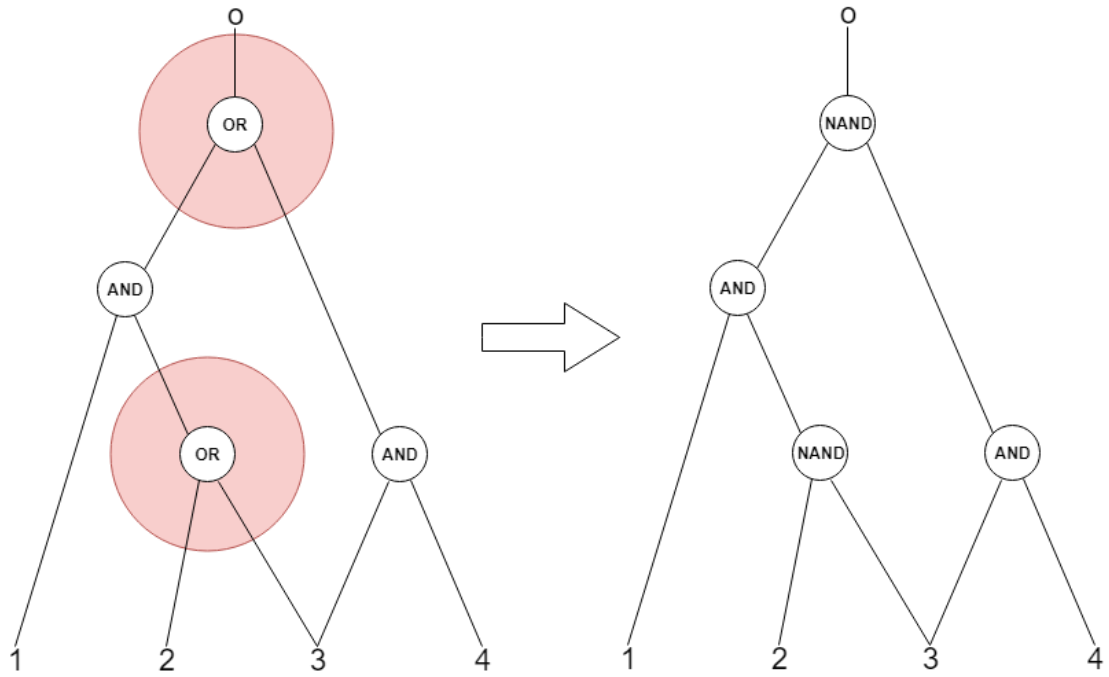

Fig. 5. a). Circuit Boolean C susceptibil atacului backtracking

Fig. 5. b). Circuit Boolean C'. Transformarea porților OR în porți $NAND$

Motivul pentru care s-a recurs la modificarea porții OR în poartă $NAND$ a fost datorită legii De Morgan și a siguranței criptografice a funcționalității porții AND. Funcționalitatea porții $NAND$ este asemănătoare cu cea a porții AND, logica noii porți fiind discutată mai îndetaliat în următoarea secțiune 3.2.

3.3 Schimbarea sistemul criptografic

Existența unei noi porți aduce cu ea schimbări atât la nivelul fizic al circuitului Boolean, cât și la nivelul algoritmilor criptografici. Astfel algoritmii $Share$ și $Recons$ necesită niște completări în conformitate cu logica porții $NAND$.

$\underline{Share(y, C)}$:

1. Toate porțile circuitului C sunt nemarcate
2. $S(o) = (y)$
3. Dacă $(w_1, w_2, AND, W = (W_1, \dots, W_k))$ este o poartă AND nemarcată și $S(W_i) = L_i$ cu $1 \leq i \leq k$, atunci:
    a. pentru fiecare element $l \in L_i$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează x_l^2 astfel încât $l = (x_l^1 + x_l^2) \bmod p$, cu $1 \leq i \leq k$
    b. se calculează
    $$WL_1 = ((x_l^1 \mid pentru\ fiecare\ l \in L_i) \mid 1 \leq i \leq k)$$
    $$WL_2 = ((x_l^2 \mid pentru\ fiecare\ l \in L_i) \mid 1 \leq i \leq k)$$
    c. se asignează $S(w_1) = WL_1$ și $S(w_2) = WL_2$
4. Dacă $(w_1, w_2, NAND, W = (W_1, \dots, W_k))$ este o poartă $NAND$ nemarcată și $S(W_i) = L_i$ cu $1 \leq i \leq k$, atunci:
    a. pentru fiecare element $l \in L_i$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează x_l^2 astfel încât $l = -x_l^1 - x_l^2 \bmod p$, cu $1 \leq i \leq k$
    b. se calculează
    $$WL_1 = ((-x_l^1 \mid pentru\ fiecare\ l \in L_i) \mid 1 \leq i \leq k)$$
    $$WL_2 = ((-x_l^2 \mid pentru\ fiecare\ l \in L_i) \mid 1 \leq i \leq k)$$
    c. se asignează $S(w_1) = WL_1$ și $S(w_2) = WL_2$

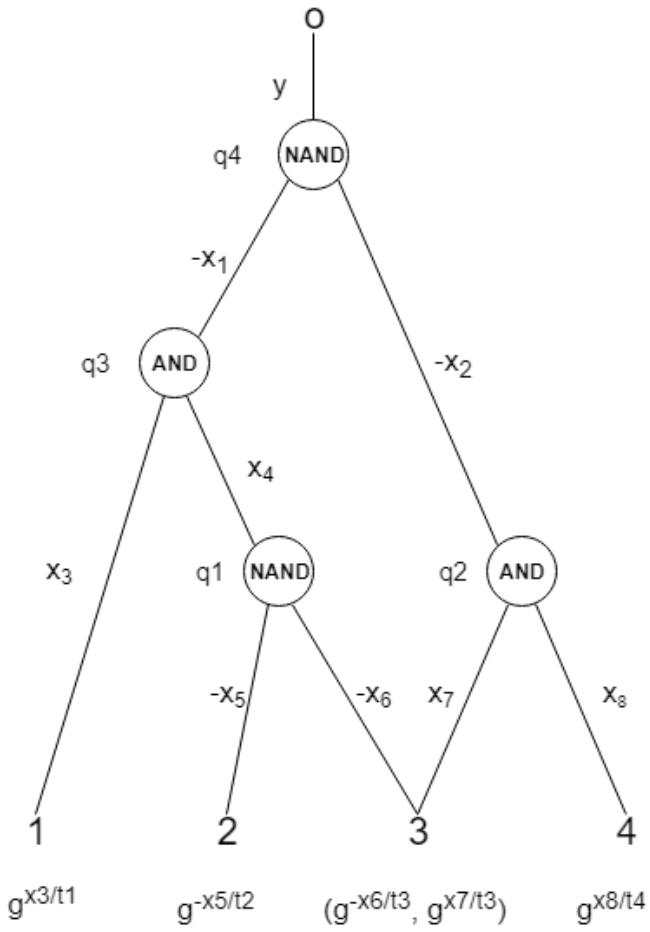

Fig. 6. Algoritmul $Share$ aplicat pe circuitul Boolean C

În continuare se va prezenta algoritmul $Share$ pe circuitul Boolean din Fig. 6.

Se primește parametrul de intrare $y \in \mathbb{Z}_p$. Conform algoritmului de criptare $KeyGen$, pentru $MSK = (y, t_1, \dots, t_4)$, avem:

1. Se alege aleator $s \in \mathbb{Z}_p$
2. Se apelează apoi algoritmul $Share$ în care se fac următorii pași:
    a. Pentru poarta $NAND$ notată $q4$ cu $W = (W_1)$ și $L(W_1) = \big((y)\big) = y$ avem $S(w_1) = \big((-x_1)\big) = -x_1$ și $S(w_2) = \big((-x_2)\big) = -x_2$, unde x_1 este generat aleator uniform din $\mathbb{Z}_p$ și x_2 se calculează astfel $-x_1 - x_2 = y \; mod \; p$.
    b. Pentru poarta AND notată $q3$ cu $W = (W_1)$ și $L(W_1) = \big((-x_1)\big) = -x_1$ avem $S(w_1) = ((x_3)) = x_3$ și $S(w_2) = ((x_4)) = x_4$, unde x_3 este generat aleator uniform din $\mathbb{Z}_p$ și x_3 se calculează astfel încât $x_3 + x_4 = -x_1 \; mod \; p$.

c. Pentru poarta $NAND$ notată $q1$ cu $W = (W_1)$ și $L(W_1) = ((x_4)) = x_4$ avem $S(w_1) = ((-x_5)) = -x_5$ și $S(w_2) = ((-x_6)) = -x_6$, unde x_5 este generat aleator uniform din $\mathbb{Z}_p$ și x_6 se calculează astfel încât $-x_5 - x_6 = x_4 \ mod \ p$.

d. Pentru poarta AND notată $q2$ cu $W = (W_1)$ și $L(W_1) = \big((-x_2)\big) = -x_2$ avem $S(w_1) = ((x_7)) = x_7$ și $S(w_2) = ((x_8)) = x_8$, unde x_7 este generat aleator uniform din $\mathbb{Z}_p$ și x_8 se calculează astfel încât $x_7 + x_8 = -x_2 \ mod \ p$.

3. Se returnează cheia de decriptare:

$$D = ((g^{\frac{x_3}{t_1}}), (g^{\frac{-x_5}{t_2}}), \Big((g^{-\frac{x_6}{t_3}}), (g^{\frac{x_7}{t_3}})\Big), (g^{\frac{x_8}{t_4}})) = (g^{\frac{x_3}{t_1}}, g^{\frac{-x_5}{t_2}}, \Big(g^{-\frac{x_6}{t_3}}, g^{\frac{x_7}{t_3}}\Big), g^{\frac{x_8}{t_4}}).$$

$\underline{Recon(C, V, g^s)}$: Se notează cu $S_i(w)$ lista de valori de pe poziția i din $S(w)$.

1. $R(i) = V(i)$, pentru toate elementele $i \in U$
2. Dacă $(w_1, w_2, AND, W = (W_1, \dots, W_k))$ este o poartă AND nemarcată cu $R(w_1)$ și $R(w_2)$ definite atunci se asignează

$$R(W_i, j) = R\big(w_{1_i}, j\big) \cdot R(w_{2_i}, j), \text{ cu } 1 \leq i \leq k \text{ și } 1 \leq j \leq |S_i(w_1)|$$

3. Dacă $(w_1, w_2, NAND, w)$ este o poartă $NAND$ nemarcată cu $R(w_1)$ și $R(w_2)$ definite atunci se asignează

$$R(W_i, j) = R\big(w_{1_i}, j\big) \cdot R(w_{2_i}, j), \text{ cu } 1 \leq i \leq k \text{ și } 1 \leq j \leq |S_i(w_1)|$$

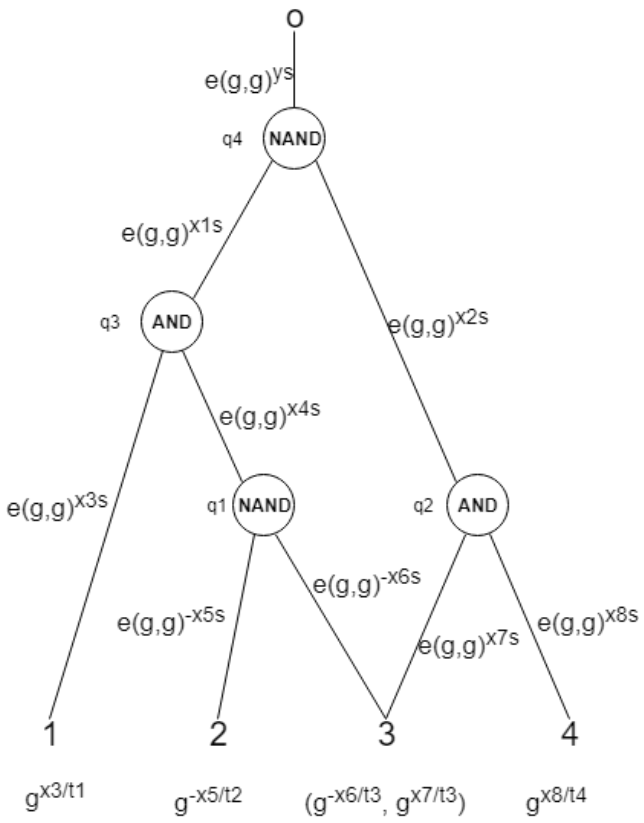

Fig. 7. Algoritmul $Recon$ aplicat pe circuitul Boolean C

În continuare se va prezenta algoritmul $Recon$ pe circuitul Boolean din Fig. 7.

Se primește cheia de decriptare $D = (g^{\frac{x_3}{t_1}}, g^{\frac{-x_5}{t_2}}, \left(g^{-\frac{x_6}{t_3}}, g^{\frac{x_7}{t_3}}\right), g^{\frac{x_8}{t_4}})$. Conform algoritmului de decriptare $Decrypt$, pașii de calcul a reconstrucției pentru setul de parametrii

$$E = (A = \{1,2,3,4\}, E' = me(g,g)^{ys}, (g^{t_1 s}, g^{t_2 s}, g^{t_3 s}, g^{t_4 s}), g^s)$$

sunt:

1. Se calculează:

$$V_A(i,j) = e\big(E_i, D(i,j)\big) = e\left(g^{t_i s}, g^{\frac{S(i,j)}{t_i}}\right) = e(g,g)^{S(i,j)s}, i \in A, 1 \le j \le |S(i)|$$

2. V_A are următoarele valori:

$$V_A = \{e(g,g)^{x_3 s}, e(g,g)^{-x_5 s}, (e(g,g)^{-x_6 s}, e(g,g)^{x_7 s}), e(g,g)^{x_8 s}\}$$

3. Se apelează apoi algoritmul $Recon$ pe parametrii de intrare C, V_A și g^s, unde C este circuitul Boolean de mai sus. În $Recon$ avem:

a. $R = \{e(g,g)^{x_3 s}, e(g,g)^{-x_5 s}, (e(g,g)^{-x_6 s}, e(g,g)^{x_7 s}), e(g,g)^{x_8 s}\}$
b. Pentru poarta $NAND$ notată $q1$ avem $e(g,g)^{-x_5 s} \cdot e(g,g)^{-x_6 s} = e(g,g)^{-x_5 s + (-x_6)s} = e(g,g)^{-(x_5 + x_6)s}$. Știm din exemplul de mai sus că în $Share$ $-(x_5 + x_6) = x_4$ și deci $e(g,g)^{-(x_5+x_6)s} = e(g,g)^{x_4 s}$.
c. Pentru poarta AND notată $q2$ avem $e(g,g)^{x_7 s} \cdot e(g,g)^{x_8 s} = e(g,g)^{(x_7+x_8)s}$. Știm de mai sus că $x_7 + x_8 = x_2$ și deci $e(g,g)^{(x_7+x_8)s} = e(g,g)^{x_2 s}$.
d. Pentru poarta AND notată $q3$ avem $e(g,g)^{x_3 s} \cdot e(g,g)^{x_4 s} = e(g,g)^{(x_3+x_4)s}$. Știm de mai sus că $x_3 + x_4 = x_1$ și deci $e(g,g)^{(x_3+x_4)s} = e(g,g)^{x_1 s}$.
e. Pentru poarta $NAND$ notată $q4$ avem $e(g,g)^{x_1 s} \cdot e(g,g)^{x_2 s} = e(g,g)^{(x_1+x_2)s}$. Știm de mai sus că $x_1 + x_2 = y$ și deci $e(g,g)^{(x_1+x_2)s} = e(g,g)^{ys}$
f. $Recon$ returnează structura descrisă mai sus.

4. Parametrul R primește structura returnată de $Recon$.
5. Mesajul se decriptează astfel

$$\frac{E'}{R(o,1)} = \frac{me(g,g)^{ys}}{e(g,g)^{ys}} = m$$

După cum se poate observa, odată cu eliminarea porții FO dispare lista P de valori din schema criptografică. Calculul porții $NAND$ în $Recon$ se rezumă la aceeași operație multiplicativă ca în cazul porții AND.

În final, noua schemă KP-ABE pe circuite Booleene generale cu forme biliniare este definită astfel:

$\underline{Setup(\lambda, n)}$:

1. Se folosește parametrul λ pentru a alege un număr prim p, două grupuri multiplicative G_1, G_2 de ordin p, un generator g din G_1 și o formă biliniară $e: G_1 \times G_1 \to G_2$
2. Se definește setul de atribute $U = \{1,2, \dots, n\}$
3. Se alege $y \in \mathbb{Z}_p$
4. Pentru fiecare $i \in U$ se definește $t_i \in \mathbb{Z}_p$
5. Se returnează

$$PP = (p, G_1, G_2, g, e, n, Y = e(g,g)^y, (T_i = g^{t_i} | i \in U))$$

$$MSK = (y, t_1, \dots, t_n)$$

$\underline{Encrypt(m, A, PP)}$:

1. Se generează aleator $s \in \mathbb{Z}_p$
2. Se returnează

$$E = (A, E' = mY^s, (E_i = T_i = g^{t_i s}), g^s)$$

$\underline{KeyGen(C, MSK)}$:

1. Se apelează $Share(y, C)$ și se reține valoarea returnată în S
2. Se returnează $D = (D(i) | i \in U)$, unde $D(i) = \left(g^{\frac{S(i,j)}{t_i}} \middle| 1 \le j \le |S(i)| \right)$ pentru fiecare $i \in U$

$\underline{Decrypt(E, D)}$:

1. Se calculează $V_A = (V_A(i) | i \in U)$ unde $V_A(i, j) = e\big(E_i, D(i, j)\big) = e\left(g^{t_i s}, g^{\frac{S(i,j)}{t_i}}\right) = e(g, g)^{S(i,j)s}$, pentru fiecare $i \in A$ și $1 \le j \le |S(i)|$, și $V_A(i)$ este o listă de $|S(i)|$ simboluri $\perp$, pentru fiecare $i \in U - A$
2. Se apelează funcția $Recon$ și se reține în variabila R: $R = Recon(C, V_A, g^s)$
3. Mesajul se decriptează $m = E'/R(o, 1)$

3.4 Securitatea soluției

Se va începe prin a se arăta că noua schema criptografică este sigură la atacul backtracking. Se cunoaște faptul că problema atacului se poate discuta doar atunci când există porți OR conectate la un fanout mai mare de unu. În acest model se renunță la ele și se înlocuiesc cu porți $NAND$. Pentru a demonstra că noua construcție previne atacul backtracking se va detalia algoritmul $Share$ pe noile porți $NAND$.

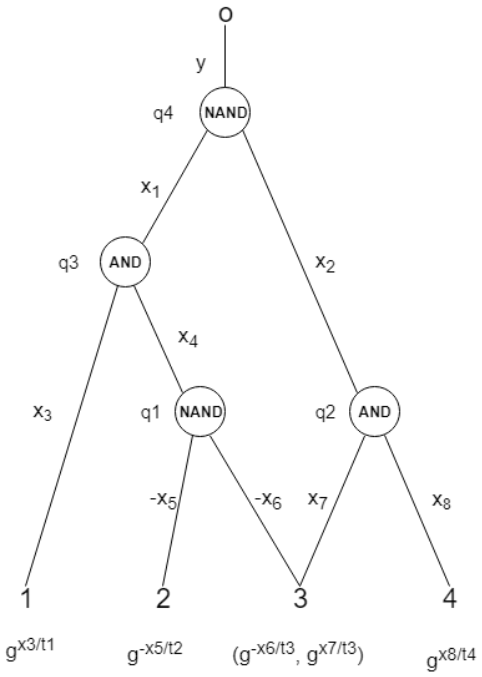

Fig. 8. a). Circuit Boolean C

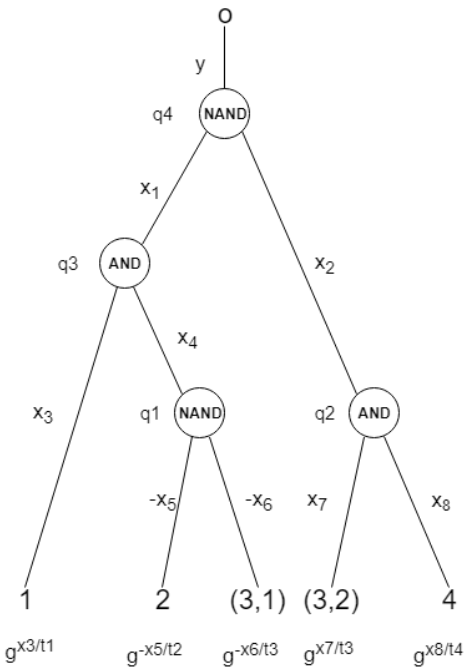

Fig. 8. b). Arborele de acces T_C

În cazul circuitului Boolean din Fig. 8. a)., funcționalitatea porții $NAND$ impune o generare aleatoare pentru $x_3 \in \mathbb{Z}_p$, iar x_4 este definit astfel încât $-x_3 - x_4 = x_2 \; mod \; p$. Dacă s-ar fi păstrat poarta OR atunci $x_3 = x_4 = x_2$, pe când în această construcție se dorește ca valorile să fie diferite.

Demonstrația securității porților AND la atacul backtracking este echivalentă cu cea a porților AND. Se realizează partajare de secrete, valorile rezultate fiind termeni ai adunării modulară.

Pentru o mai bună înțelegere a demonstrației, se construiește arborele de acces al circuitului Boolean C din fig. 8. b). prin multiplicarea firele circuitului în funcție de numărul de drumuri până la output. Se deosebesc drumurile adăugându-se un identificator. Astfel pentru atributul 3 avem separarea firului în $(3,1)$ și $(3,2)$ deoarece există două drumuri spre output.

Se poate observa atât din funcționalitatea porții $NAND$, cât și din arborele de acces T_C că atacul backtracking nu poate avea loc.

A doua parte din această secțiune este demonstrarea securității modelul cu ajutorul ipotezei problemei decizionale biliniare Diffie-Hellman.

Fie p un număr impar prim, G_1 și G_2 două grupuri multiplicative ciclice de ordin p și e o formă biliniară definită astfel $e: G_1 \times G_1 \to G_2$.

Se prezintă jocul criptografic dintre adversarul A cu timp polinomial de rezolvare și oracolul B. Oracolul primește o instanță a ipotezei problemei decizionale biliniare Diffie-Hellman (g^a, g^b, g^c, Z_v) cu Z_v ales de către B, unde $Z_v \leftarrow \{Z_0, Z_1\}$, $Z_0 = e(g,g)^{abc}$ și $Z_1 = e(g,g)^z$, g generator al grupului multiplicativ ciclic G_1și $a, b, c, z \in \mathbb{Z}_p$.

Inițializare. Fie M mulțimea nenulă de atribute aleasă de adversarul A.

Preliminarii. Oracolul B alege aleator $r_i \in \mathbb{Z}_p$ pentru toate atributele $i \in U$ și calculează $Y = e(g^a, g^b) = e(g,g)^{ab}$ și $T_i = g^{t_i}$ pentru toate atributele $i \in U$, unde

$$t_i = \begin{cases} r_i, & i \in M \\ br_i, & altfel \end{cases}$$

și afișează parametrii publici

$$PP = (p, G_1, G_2, g, e, n, Y, (T_i | i \in U))$$

Prima parte. Adversarul primește acces la oracolul de generare de chei de decriptare pentru toate circuitele C cu $C(M) = 0$. Din perspectiva adversarului, partajarea de secrete și distribuția cheilor de decriptare trebuie să arate ca în schema originală. Reconstrucția cu ajutorul funcției $Recon$ trebuie să returneze $e(g,g)^{abc}$.

Odată primită cererea, se convertește circuitul în arbore de acces și se începe procesul de partajare de secrete:

1. $S(o) = g^a$
2. Pentru $(w_1, w_2, AND, W = (W_1, \dots, W_k))$ cu $S(W_j) = L_j$, unde $1 \le j \le k$ atunci pentru fiecare W_i avem:
    a. Dacă $C_{W_i}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $x_l^2 = (l - x_l^1) \bmod p$. Se definesc

$$L_1^j = (x_l^1 | pentru\ fiecare\ l \in L_j)$$
$$L_2^j = (x_l^2 | pentru\ fiecare\ l \in L_j).$$
$$S(w_1) = \left((L_1^j) | 1 \le j \le k\right)$$
$$S(w_2) = ((L_2^j) | 1 \le j \le k).$$

b. Dacă $C_{W_i}(M) = 0 = C_{w_2}(M)$ și $C_{w_1}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $g^{x_l^2} = l/g^{x_l^1}$. Se definesc:

$$L_1^j = (x_l^1 | pentru\ fiecare\ l \in L_j)$$

$$L_2^j = (g^{x_l^2} | pentru\ fiecare\ l \in L_j).$$

$$S(w_1) = ((L_1^j) | 1 \leq j \leq k)$$

$$S(w_2) = ((L_2^j) | 1 \leq j \leq k).$$

c. Dacă $C_{W_i}(M) = 0 = C_{w_1}(M)$ și $C_{w_2}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^2 \in \mathbb{Z}_p$ și se calculează $g^{x_l^1} = l/g^{x_l^2}$. Se definesc:

$$L_1^j = (g^{x_l^1} | pentru\ fiecare\ l \in L_j)$$

$$L_2^j = (x_l^2 | pentru\ fiecare\ l \in L_j).$$

$$S(w_1) = \left((L_1^j) | 1 \leq j \leq k\right)$$

$$S(w_2) = ((L_2^j) | 1 \leq j \leq k).$$

d. Dacă $C_{W_i}(M) = C_{w_1}(M) = C_{w_2}(M) = 0$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $g^{x_l^2} = l/g^{x_l^1}$. Se definesc:

$$L_1^j = (g^{x_l^1} | pentru\ fiecare\ l \in L_j)$$

$$L_2^j = (g^{x_l^2} | pentru\ fiecare\ l \in L_j).$$

$$S(w_1) = ((L_1^j) | 1 \leq j \leq k)$$

$$S(w_2) = ((L_2^j) | 1 \leq j \leq k).$$

3. Pentru $(w_1, w_2, NAND, W = (W_1, \dots, W_k))$ cu $S(W_j) = L_j$, unde $1 \leq j \leq k$ atunci pentru fiecare W_i avem:

a. Dacă $C_{W_i}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $x_l^2 = (x_l^1 - l)\ mod\ p$. Se definesc:

$$L_1^j = (-x_l^1 | pentru\ fiecare\ l \in L_j)$$

$$L_2^j = (-x_l^2 | pentru\ fiecare\ l \in L_j).$$

$S(w_1) = ((L_1^j) | 1 \leq j \leq k)$ și $S(w_2) = ((L_2^j) | 1 \leq j \leq k)$.

b. Dacă $C_{W_i}(M) = 0 = C_{w_2}(M)$ și $C_{w_1}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $g^{-x_l^2} = l/g^{-x_l^1}$. Se definesc:

$$L_1^j = (-x_l^1 | pentru\ fiecare\ l \in L_j)$$
$$L_2^j = (g^{-x_l^2} | pentru\ fiecare\ l \in L_j).$$
$$S(w_1) = ((L_1^j) | 1 \le j \le k)$$
$$S(w_2) = ((L_2^j) | 1 \le j \le k).$$

c. Dacă $C_{W_i}(M) = 0 = C_{w_1}(M)$ și $C_{w_2}(M) = 1$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^2 \in \mathbb{Z}_p$ și se calculează $g^{-x_l^1} = l/g^{-x_l^2}$. Se definesc:

$$L_1^j = (g^{-x_l^1} | pentru\ fiecare\ l \in L_j)$$
$$L_2^j = (-x_l^2 | pentru\ fiecare\ l \in L_j).$$
$$S(w_1) = ((L_1^j) | 1 \le j \le k)$$
$$S(w_2) = ((L_2^j) | 1 \le j \le k).$$

d. Dacă $C_{W_i}(M) = C_{w_1}(M) = C_{w_2}(M) = 0$ atunci pentru fiecare $l \in L_j$ se alege aleator uniform $x_l^1 \in \mathbb{Z}_p$ și se calculează $g^{-x_l^2} = l/g^{-x_l^1}$. Se definesc:

$$L_1^j = (g^{-x_l^1} | pentru\ fiecare\ l \in L_j)$$
$$L_2^j = (g^{-x_l^2} | pentru\ fiecare\ l \in L_j).$$
$$S(w_1) = \left((L_1^j) | 1 \le j \le k\right)$$
$$S(w_2) = ((L_2^j) | 1 \le j \le k).$$

B va returna adversarului A cheia de decriptare $D = (D(i) | i \in U)$, unde

$$D(i) = \begin{cases} ((g^b)^{S_i(j)/r_i} \mid 1 \le j \le |S(i)|), i \in M \\ (S_i(j)^{\frac{1}{r_i}} \mid 1 \le j \le |S(i)|), altfel \end{cases}, \forall i \in U.$$

Jocul de securitate. Adversarul A alege două mesaje de aceeași lungime m_0 și m_1 și le trimite lui B, care criptează m_u cu Z_v, unde $Z_v \leftarrow \{Z_0 = e(g,g)^{abc}, Z_1 = e(g,g)^z\}$ și îl trimite înapoi adversarului. Setul de parametrii pentru mesajul criptat este:

$$E = (M, E' = m_u Z_v, \{E_i = T_i^c = g^{cr_i}\}_{i \in M}).$$

Dacă $v = 0$, atunci E este o criptare validă a mesajului m_u, altfel E' este un element aleator din G_2.

A doua parte. Adversarul primește încă o dată acces la oracolul generării cheii de decriptare sub aceleași criterii ca în **prima parte**.

Presupunerea. Fie u' presupunerea adversarului A. Dacă $u' = u$ atunci B returnează $v' = 0$, altfel $v' = 1$.

Astfel se calculează avantajul lui B.

$$P(v' = v) - \frac{1}{2} = P(v' = 0|v = 0) \cdot P(v = 0) + P(v' = 1|v = 1) \cdot P(v = 1) - \frac{1}{2}$$

Știm că v este ales aleator uniform din $\{0,1\}$, deci $P(v = 0) = P(v = 1) = 1/2$. De asemenea, observăm că

$$P(v' = v|v = 0) = P(u' = u|v = 0) = \frac{1}{2} + \alpha$$

$$P(v' = v|v = 1) = P(u' \neq u|v = 1) = 1/2.$$

În concluzie, avantajul lui B este

$$P(v' = v) - \frac{1}{2} = \left(\frac{1}{2} + \alpha\right) \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} - \frac{1}{2} = \frac{1}{4} + \frac{\alpha}{2} + \frac{1}{4} - \frac{1}{2} = \frac{\alpha}{2}$$

, demonstrând astfel că modelul criptografic este sigur.

3.5 Complexitatea noii scheme

În această secțiune se va prezenta complexitatea modelului propus în comparație cu celelalte două abordări descrise mai sus în Capitolul 2.

Pentru a putea discuta despre complexitatea soluției trebuie mai întâi detaliat algoritmul $Share$. Conform funcționalității acestuia, algoritmul partajează secrete prin fire de la output la input astfel:

- Poarta (w_1, w_2, AND, W) trimite mai departe câte o nouă valoare pentru w_1 și pentru w_2 pentru fiecare element din $S(W)$
- Poarta $(w_1, w_2, NAND, W)$ trimite de asemenea mai departe câte o nouă valoare pentru w_1 și pentru w_2 pentru fiecare element din $S(W)$

În concluzie, algoritmul $Share$ trimite pentru fiecare element din firele circuitului Boolean câte două valori și deci numărul de valori depinde strict de dimensiunea circuitului. S-a observat că în cel mai rău caz numărul de elemente crește exponențial în funcție de numărul de nivele ale circuitului, lucru ce era de așteptat datorită funcționalității algoritmului. Prin urmare putem discuta despre cazurile de complexitate astfel:

I. Cazul cel mai favorabil este acela în care există o construcție care să genereze exact n elemente. Cel mai simplu și evident exemplu este atunci când n poate fi scris sub forma $2^k, k \in \mathbb{N}$. În această situație toate firele circuitului Boolean primesc strict o valoare și deci pe nivelul de atributelor se vor găsi n elemente care construiesc cheia de decriptare.

II. Cazul cel mai puțin favorabil este acela când desfășurarea circuitului Boolean se face pe m nivele unde firele comunică toate cu poarta învecinată. Funcționalitatea algoritmului $Share$ impune creerea de noi valori pentru fiecare nou input primit prin fir și creează o listă nouă cu aceste noi valori pentru fiecare fir în parte, creând astfel de două ori numărul de valori de pe nivel anterior. Cheia de decriptare ajunge să aibă ca număr de elemente 2^m.

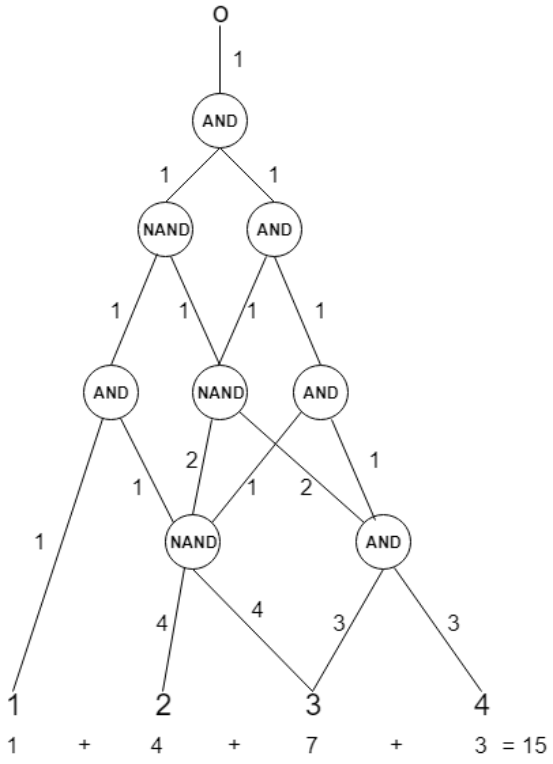

Fig. 9. Exemplu circuit Boolean C. Număr de valori pentru cheia de decriptare

Fie n numărul de atribute și r numărul de porți cu fanout de dimensiune j. În continuare, se prezintă complexitățile celorlalte scheme criptografice în comparație cu cea a soluției propuse.

Caz	Soluția [1]	Soluția [4]	Soluția propusă
Favorabil	$nj + n + r(j-1)$	$n + r(j-1)$	n
Cel mai puțin favorabil	$nj + n + j^r$	$n + j^r$	2^m

Din tabel se poate observa că numărul de elemente din această schemă criptografică nu depinde de dimensiunea fanout-ului și nici de numărul de fanout-uri. Schema criptografică depinde strict de numărul de niveluri ale circuitului Boolean.

Pentru evidențierea diferențelor de complexitate, se construiesc soluțiile modelelor [1] și [4] pe circuitul Boolean C din Fig. 9.

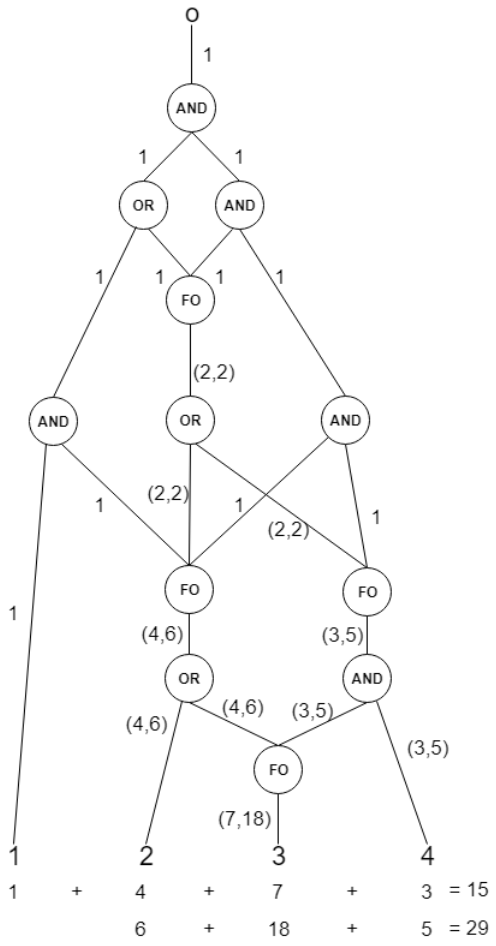

Fig. 10. Exemplu circuit Boolean C. Număr de valori pentru cheia de decriptare pentru modelul din lucrarea [1].

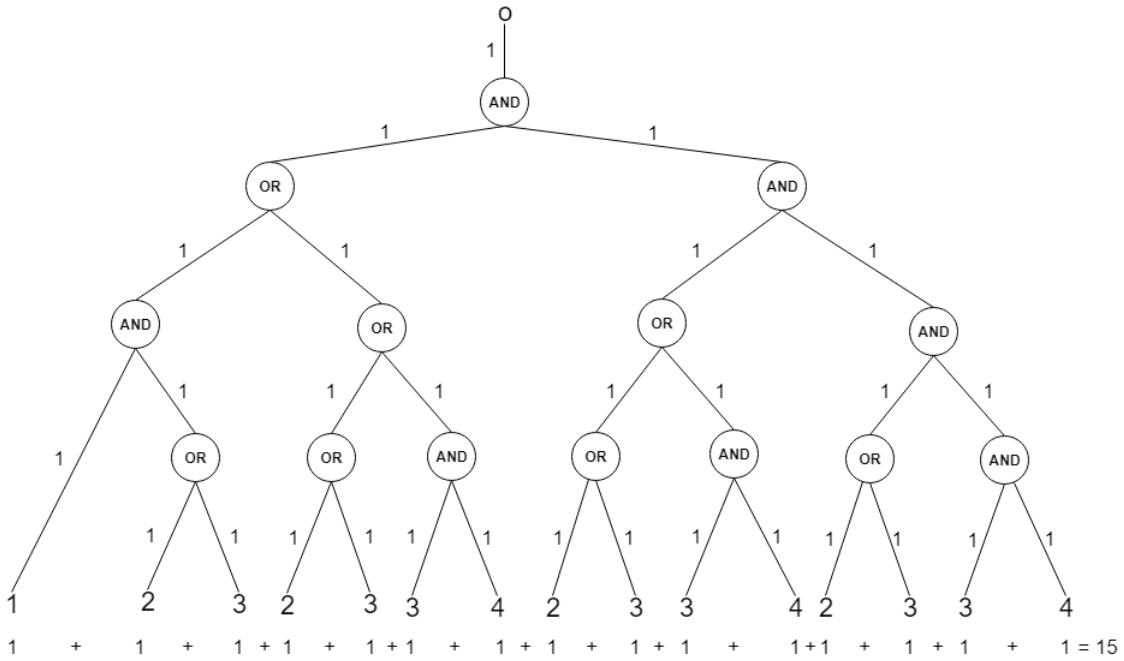

Fig. 11. Exemplu circuit Boolean C. Număr de valori pentru cheia de decriptare pentru modelul din lucrarea [4].

După cum se poate observa din imaginile de mai sus, schemele criptografice definite în lucrările [1] și [4] au propriile lor dezavantaje. În cazul întâlnit în Fig. 10. modelul generează pe lângă cele cincisprezece elemente de bază, încă douăzeci și nouă de elemente secundare necesare datorită funcționalității porții FO, pe când în Fig. 11. nu se mai produc noi variabile dar dimensiunea circuitului Boolean crește.

3.6 Concluzii

Noua schemă criptografică este sigură la atacul backtracking prin funcționalitatea porților AND și $NAND$. De asemenea, aceasta este o schemă criptografică sigură datorită presupunerii problemei decizionale biliniare Diffie-Hellman aplicată mai sus.

Complexitatea modelului depinde strict de numărul de nivele al circuitului Boolean și nu interesează cât de complex este acesta. Funcționalitatea porților permite păstrarea structurii compacte a circuitului, evitându-se astfel adăugarea de noi porți sau extinderea acestuia. Pe lângă aceasta, eliminarea porților FO reduce numărul de valori reținute și de calcule necesare obținerii acestora.

În concluzie, noua schemă criptografică este o schemă sigură și mai eficientă decât construcțiile precedente.

Concluzii finale

Lucrarea „Schema KP-ABE pentru circuite Booleene generale cu forme biliniare” reușește să suprindă o abordare validă și eficientă a generalizării schemelor KP-ABE în contextul circuitelor Booleene.

Am reușit prin îmbinarea noțiunilor teoretice criptografice și a circuitelor Booleene să creez un model criptografic sigur din punct de vedere al securității și care păstrează complexitatea circuitului Boolean, fără a-l extinde, prin transformări ale porților.

Pentru viitor această soluție ar putea fi îmbunătățită. După cum se poate observa, modelul propus are la bază o trasformare totală generală, în care nu se iau în considerare construcții speciale. O abordare care ar putea compacta circuitul Boolean, ocupând prin urmare mai puțin spațiu prin reducerea numărului de porți, este trecerea construcțiilor formate din disjuncții de conjuncții în construcții (k,n) precum în Fig. 12.

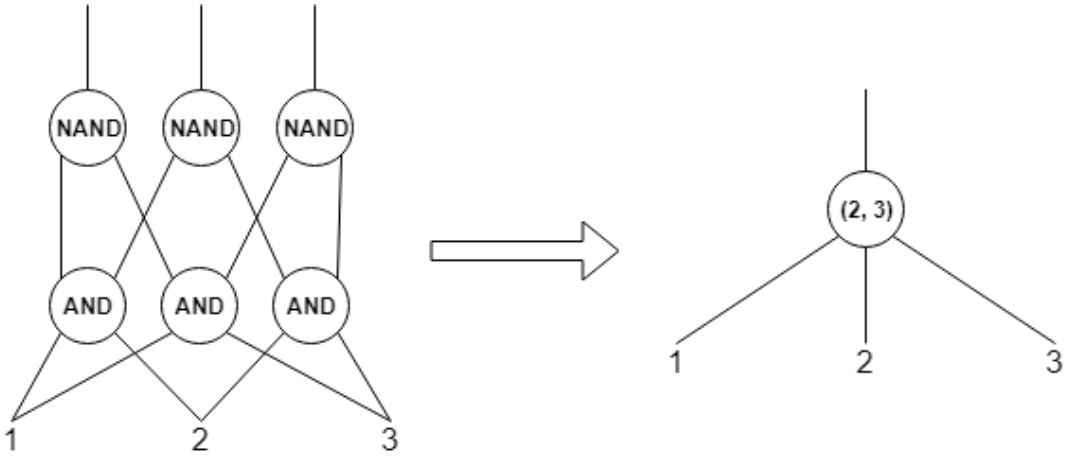

Fig. 12. Exemplu de poartă (k,n)

Această soluție ar putea avea la bază schema Shamir sau schema Mignotte pentru partajarea secretelor. Mai multe detalii despre aceste modele de partajare de secrete sunt discutate în Capitolul 1 la secțiunea 1.1 de noțiuni criptografice.

Pentru fiecare secret introdus prin firul de ieșire al porții (k, n) se va genera câte un secret partajat pentru fiecare fir de intrare după funcționalitatea modelului de partajare

care va sta la baza porții. Intuitiv, numărul de elemente din cheia de decriptare va stagna, însă spațiul fizic ocupat de circuitul Boolean va scădea semnificativ.

Din punct de vedere al eficienței, se cunoaște că ar putea apărea niște impedimente în cazul calculului inversei modulare, operație de bază în interpolarea polinomială din schema Shamir, însă o soluție pentru această problemă ar putea fi alegea unui număr prim p de forma $2^k - 1$, unde k este un număr natural mai mare sau egal cu 2, cunoscut și sub numele de număr Mersenne. Numerele Mersenne sunt folosite în calculul inverselor modulare datorită formei lor ce permit ca operația să fie eficientă.

Bibliografie

[1] Ferucio Laurențiu Țiplea și Constantin Cătălin Drăgan. Key-policy Attribute-based Encryption for Boolean Circuits from Bilinear Maps. În BalkanCryptSec 2014, Istanbul, Turcia, 16-17 Octombrie, 2014, paginile 175-193, LNCS 9024

[2] John Bethencourt, Amit Sahai și Brent Waters. Ciphertext-policy attribute based encryption. În IEEE Symposium, Security and Privacy, 2007, paginile 321–334. IEEE Computer Society, 2007

[3] Vipul Goyal, Omkant Pandey, Amit Sahai și Brent Waters. Attribute-based encryption for fine-grained access control of encypted data. În ACM Conference on Computer and Communications Security, paginile 89–98. ACM, 2006. Preprint în IACR ePrint 2006/309

[4] Peng Hu și Haiying Gao. A Key-Policy Attribute-based Encryption Scheme for General Circuit from Bilinear Maps. În International Journal of Network Security, Vol.19, Nr.5, paginile 704-710, Septembrie 2017

[5] Sanjam Garg, Craig Gentry, Shai Halevi, Amit Sahai și Brent Waters. Attribute based encryption for circuits from multilinear maps. In Ran Canetti și JuanA. Garay, editori, Advances in Cryptology CRYPTO 2013, volumul 8043 de Lecture Notes in Computer Science, paginile 479–499. Springer Berlin Heidelberg, 2013. Preprint în IACR ePrint 2013/128