UNIVERSITATEA DIN BUCUREȘTI

FACULTATEA DE MATEMATICĂ ȘI INFORMATICĂ

SPECIALIZAREA INFORMATICĂ

Lucrare de licență

PROTOCOALE PENTRU LICITAȚII SECURIZATE FOLOSIND ANGAJAMENTE DE TIMP

Absolvent
Apostol Alin-Constantin

Coordonator științific
Conf.dr. Ruxandra F. Olimid

București, iunie 2025

Rezumat

Lucrarea prezintă noțiunea de sisteme pentru angajamente de timp (*timed commitment schemes*) prin care, o posibilă acțiune de deschidere (*opening phase*) forțată a unui angajament permite (în urma unui efort computațional considerabil depinzând de un parametru de timp T) accesarea valorii pe care o conține acesta, fără intervenția părții care a realizat angajamentul. Această noțiune este o extindere a sistemelor pentru angajament standard, în care valoarea poate fi aflată doar prin intermediul valorii de deschidere pe care o deține persoana care s-a angajat. Licitațiile în care ofertele sunt secrete sunt o aplicație importantă pentru sistemele pentru angajamente de timp. Lucrarea prezintă o implementare concretă și personală a unui astfel de sistem, precum și o simulare a unui protocol care asigură onestitatea în cadrul unei licitații de acest tip.

Abstract

The thesis presents the notion of timed commitment schemes, whereby a possible forced opening action of a commitment allows (after considerable computational effort depending on a time parameter T) access to the value that it contains, without the intervention of the committer. This scheme is an extension of the standard commitment scheme, in which the value can be retrieved only through the opening value held by the party who made the commitment. Sealed-bid auctions are an important application of such schemes. The thesis presents a personal implementation of a timed commitment scheme and a simulation of a honesty-preserving protocol for this type of auctions.

Cuprins

1 Introducere **4**
1.1 Motivație 4
1.2 Prezentare generală 4
1.3 Contribuția personală 5
1.4 Structura lucrării 5

2 Preliminarii **6**
2.1 Tipuri de licitații 6
2.1.1 Licitații deschise 6
2.1.2 Licitații cu oferte secrete 6
2.2 Sisteme de angajament 7

3 Sisteme de angajament **9**
3.1 Sisteme de angajament Pedersen 9
3.2 Sisteme pentru angajamente de timp - Boneh-Naor 11
3.2.1 Inițializarea sistemului 12
3.2.2 Etapa de angajament 12
3.2.3 Etapa de deschidere 15
3.2.4 Etapa de deschidere forțată 15

4 Protocoale pentru licitații securizate **17**
4.1 Prezentarea protocolului 17
4.2 Simularea protocolului 19
4.3 Situația actuală a domeniului 22

5 Concluzii **23**

Bibliografie **24**

Capitolul 1

Introducere

1.1 Motivație

Licitațiile au fost și sunt o metodă răspândită de comercializare a bunurilor, fie că este vorba de licitații în care ofertele sunt anunțate public și făcute în mod secvențial (cunoscute și sub numele de *English Auctions*), fie de cele în care ofertele sunt secrete, făcute într-o singură tură (de tipul *primul preț*, de tipul *al doilea preț* etc.).

În anul 2020, o coaliție a mai multor state condusă de către procurorul general al statului Texas, Statele Unite ale Americii, Ken Paxton, a pornit o urmărire în justiție împotriva Google, în legătură cu manipularea licitațiilor de vânzare a reclamelor online [5]. Pe data de 17 aprilie 2025, Divizia Antitrust a Departamentului de Justiție din Statele Unite ale Americii a anunțat o a doua victorie în instanță împotriva Google pe această tematică [16]. În mod cert, există numeroase metode de a obține în mod ilicit un avantaj în cadrul proceselor de licitație, din punctul de vedere al vânzătorului și al participanților. Așadar, se observă astfel apariția unei nevoi de a dezvolta sisteme sigure pentru procesele de licitație.

1.2 Prezentare generală

Lucrarea mea abordează domeniul **criptografiei**, fiind disciplina care, utilizând diverse principii și metode, își propune să ascundă înțelesul semantic al datelor, să asigure integritatea acestora și să prevină utilizarea lor neautorizată [12].

Lucrarea studiază anumite concepte legate de diferite tipuri de licitație, avantajele și dezavantajele acestora din punct de vedere al rezultatului obținut, cât și din punctul de vedere al securității. De asemenea, definește și aprofundează noțiunea de sistem de angajament (eng. orig. *commitment schemes*), studiind atât sisteme clasice, cât și cele de timp (eng. orig. *timed commitment schemes*). Lucrarea analizează diverse lucrări bazate pe sisteme de angajament, construind protocoale securizate care facilitează licitațiile online.

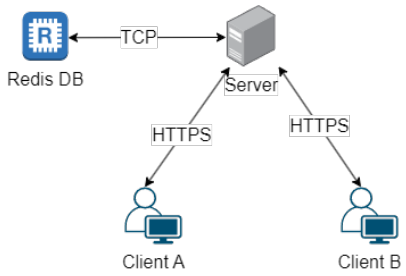

Figura 1.1: Comunicare între containere

1.3 Contribuția personală

Contribuția personală reprezintă realizarea unei simulări tehnice a unui protocol de licitație care își propune să păstreze onestitatea pe parcursul întregului proces, prezentat în cadrul lucrării [3]. Această simulare este scrisă în limbajul Python [7], folosind diverse module și biblioteci precum *ecdsa* [20], *sympy* [17], *flask* [13], *redis* [9], *requests* [15], și utilizează containere de Docker [10]. Pe scurt, se creează patru containere care reprezintă: (i) un server care găzduiește licitația și postează ofertele primite în faza finală, (ii) doi clienți care comunică doar cu serverul pentru a participa la licitație și (iii) un container de Redis pe care serverul îl folosește drept bază de date pentru a stoca date necesare protocolului de licitație. Comunicarea între părți se realizează făcând request-uri HTTPS, așa cum este prezentat și în *Figura 1.1*.

1.4 Structura lucrării

Lucrarea este structurată în următoarele capitole principale:

- **Capitolul 1. Introducere**: prezintă tema, motivația și o vedere de ansamblu a lucrării.
- **Capitolul 2. Preliminarii**: prezintă anumite noțiuni de bază legate de licitații și sisteme de angajament.
- **Capitolul 3. Sisteme de angajament**: studiază două sisteme de angajament: unul standard, Pedersen [14] și unul de timp, Boneh-Naor [3].
- **Capitolul 4. Protocoale pentru licitații**: descrie un protocol de licitație bazat pe sisteme de angajament și prezintă o implementare proprie a unei simulări a protocolului. De asemenea, expune sumar un protocol de actualitate bazat pe aceleași construcții criptografice.
- **Capitolul 5. Concluzii**: prezintă concluzii despre utilitatea sistemelor de angajament în diverse domenii și direcții viitoare de dezvoltare.

Capitolul 2

Preliminarii

2.1 Tipuri de licitații

Lucrarea prezintă două categorii de licitații: (*i*) licitații deschise (eng. original *open auctions*) și (*ii*) licitații cu ofertă secretă (eng. original *sealed-bid auctions*). Ultima dintre cele două este cea pentru care se va prezenta ulterior în cadrul lucrării un protocol care reușește să asigure onestitatea tuturor părților.

2.1.1 Licitații deschise

Licitațiile deschise reprezintă cel mai cunoscut tip de licitații practicat în vânzări. La o astfel de licitație iau parte vânzătorul, organizatorul și clienții care fac oferte pentru a achiziționa bunul. Aceasta poate începe cu un preț inițial stabilit fie de vânzător, fie de organizator, ca mai apoi clienții să facă oferte publice, cu preț în ordine crescătoare în mod secvențial. Câștigătorul licitației este cel care oferă cel mai mare preț. O asemenea licitație poartă denumirea și de *licitație engleză* (eng. original *English auctions*) [11].

Avantajul unui astfel de sistem este că rezultatul final (câștigătorul licitației) este cel care prețuiește cel mai mult bunul de vânzare. În schimb, se observă că dezavantajul este că necesită un număr mare de runde de comunicare, iar această comunicare trebuie să fie cât mai eficientă în mediul online, tocmai pentru că, în general, un astfel de proces se termină repede [11].

2.1.2 Licitații cu oferte secrete

Licitațiile cu oferte secrete sunt o modalitate răspândită de vânzare a unor bunuri în mediul online. Unul dintre cele mai răspândite tipuri de astfel de licitație sunt cele de tip *al doilea preț*, cunoscute și sub numele de licitații Vickrey [19]. În cadrul unei licitații Vickrey clasice (offline), participanții împărtășesc ofertele secrete unei părți terțe autorizate și de încredere, ca mai apoi să câștige ofertantul cu prețul cel mai mare, însă

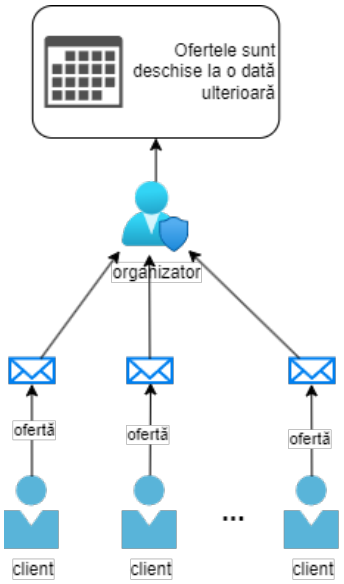

Figura 2.1: Licitație de tipul ofertă secretă

prețul de cumpărare al bunului fiind cel de-al doilea în ordine descrescătoare. William Vickrey a arătat că, într-un cadru ideal, prețul de cumpărare final este foarte apropiat de cel obținut prin licitații de tipul ofertă deschisă, în care prețul cel mai mare câștigă [18]. Bineînțeles, pot fi stabilite și alte reguli ale stabilirii prețului final de cumpărare, cum ar fi cel mai mare preț, al treilea cel mai mare preț etc. Simularea protocolului prezentat ulterior este compatibilă cu orice modalitate.

Avantajul acestui tip de licitație este faptul că necesită o singură rundă de comunicare între participanți și partea terță. În cadrul online, stabilirea unei părți terțe autorizate în care toți participanții să aibă încredere, inclusiv vânzătorul, reprezintă un impediment [18].

2.2 Sisteme de angajament

Sistemele de angajament (*commitment schemes*) sunt protocoale criptografice care permit unui participant să realizeze un angajament asupra unei valori, ascunzând valoarea inițială, cu posibilitatea de a o dezvălui mai târziu prin publicarea valorii de deschidere a angajamentului. Un astfel de sistem trebuie să ofere confidențialitatea valorii inițiale, precum și certitudinea că valoarea inițială este legată în mod direct de angajamentul făcut, adică participantul care a făcut angajamentul nu îl poate deschide pe acesta la o altă valoare decât cea pe care o conține, oferind o altă valoare de deschidere.

O modalitate de a înțelege aceste sisteme este următoarea: Alice scrie un mesaj pe un bilețel pe care îl pune într-un seif opac și îl încuie folosind un cifru; apoi, îi oferă lui Bob tot seiful, fără a dezvălui nici mesajul, nici cifrul. În acest moment, Bob este sigur că Alice nu poate modifica mesajul scris pe bilețel, iar Alice este sigură că Bob nu poate

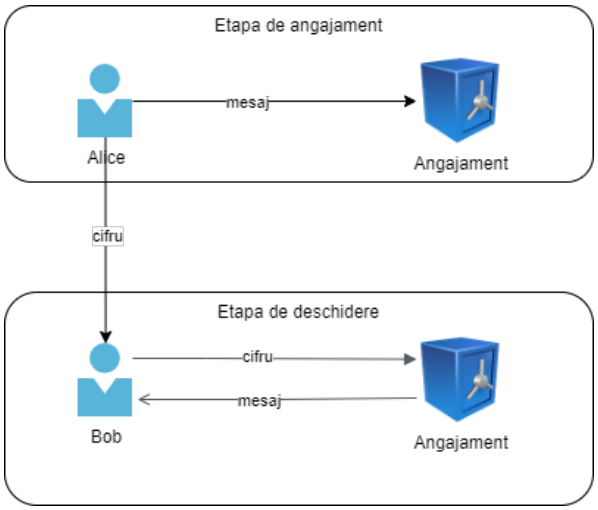

Figura 2.2: Abstractizare a unui sistem de angajament

citi mesajul până când nu îi comunică cifrul stabilit de aceasta. De asemenea, dacă Alice încearcă să trișeze și îi oferă lui Bob un alt cifru, atunci seiful nu se va deschide. Această comunicare este ilustrată în *Figura 2.2*.

Un sistem de angajament este alcătuit, în general, din două etape:

1. **Etapa de angajament**: în cadrul acestei etape, Alice creează un angajament care leagă o valoare aleasă de aceasta, apoi îl trimite către Bob. Totodată, se stabilește și valoarea de deschidere prin intermediul căreia se poate accesa valoarea conținută de angajament. Această valoare de deschidere este unică pentru angajament și poate dezvălui doar valoarea aleasă de Alice.

2. **Etapa de deschidere**: în momentul în care Alice publică valoarea de deschidere, Bob poate să citească mesajul inițial pe care Alice l-a legat de angajament.

Capitolul 3

Sisteme de angajament

3.1 Sisteme de angajament Pedersen

Un exemplu de sistem de angajament standard este cel definit de T.P. Pedersen în [14].

Fie două numere prime **mari** p și q astfel încât $q \mid (p-1)$, $\mathbb{Z}_p^*$ grupul elementelor inversabile în $\mathbb{Z}_p$, adică $\mathbb{Z}_p^* = \{x \mid cmmdc(x,p) = 1\}$. Cum p este prim, atunci avem că $\mathbb{Z}_p^* = \mathbb{Z}_p - \{1\}$, unde 1 este elementul neutru, așadar ordinul grupului $\mathbb{Z}_p^*$ este $p-1$. Fie G_q subgrup unic al lui $\mathbb{Z}_p^*$ de ordine q, adică $\mid G_q \mid = q$, și fie g un generator al acestui subgrup. A determina apartenența unui element $x \in \mathbb{Z}_p^*$ la subgrupul G_q înseamnă a verifica faptul că $x^q \equiv 1 \pmod{q}$.

Se observă că fiecare element din G_q este generator al acestuia, în afară de 1, așadar se poate defini logaritmul unui element $x \in G_q$ în baza $y \in G_q$. Acesta se notează $\log_y x$ și semnifică *puterea la care este ridicat y astfel încât se obține x*. Acest rezultat notat este mai mic decât q (ordinul grupului). Această construcție pentru angajamente de timp prezentată de Pedersen se bazează pe **problema logaritmului discret** (eng. original *discrete logarithm problem*).

Problema logaritmului discret reprezintă aflarea valorii $\log_y x$ și este de interes în domeniul criptografiei atunci când acest lucru este dificil din punct de vedere computațional. Aceasta a fost introdusă în criptografie de către W.Diffie si M.Hellman [6].

Sistemul pentru angajament introdus de Pedersen este alcătuit și el, din cele două etape [14]:

1. **Etapa de angajament**

   Pentru a realiza un angajament, Alice trebuie să aleagă două elemente $g, h \in G_q$ astfel încât să nu se cunoască valoarea $log_g h$ (problema logaritmului discret trebuie să fie dificilă). Acest lucru se poate face fie de o parte de încredere și autorizată, fie urmând un protocol de *coin-flipping* (un protocol criptografic prin care două părți generează împreună o valoare aleatoare). Cel care dorește să facă angajamentul asupra unei valori $m \in \mathbb{Z}_q$, trebuie să aleagă în mod aleator o valoare $\alpha \in \mathbb{Z}_q$, apoi

se calculează angajamentul $C = g^m h^\alpha$. Mai departe, această valoare este oferită către Bob pentru a fi *deschisă* ulterior. De asemenea, valorile $g, h \in G_q$, precum și q sunt considerați **parametrii publici** ai construcției de angajament, adică oricine are acces la aceste valori în timpul procesului.

2. **Etapa de deschidere**

   În acest moment, Bob cunoaște C, iar singura metodă prin care poate afla valoarea legată de angajament este de a primi deschiderea $\pi = (m, \alpha)$, apoi să verifice că, într-adevăr $C = g^m h^\alpha$.

Teoremă 3.1.1 *[14] Pentru orice valoare $m \in \mathbb{Z}_q$ și pentru o valoare $\alpha \in \mathbb{Z}_q$ alesă uniform aleator, angajamentul C este uniform distribuit în G_q.*

Astfel, pentru două valori $m, m' \in \mathbb{Z}_q$ cu $m \neq m'$ astfel încât $C = g^m h^t = C' = g^{m'} h^{t'}$, atunci reiese că $t \not\equiv t' \pmod q$ și că:

$$\begin{aligned} g^m h^t &= g^{m'} h^{t'} \\ \frac{g^m}{g^{m'}} &= \frac{h^{t'}}{h^t} \\ g^{m-m'} &= h^{t'-t} \\ \log_g g^{m-m'} &= \log_g h^{t'-t} \\ (m-m') \log_g g &= (t'-t) \log_g h \\ m - m' &= (t'-t) \log_g h \\ \log_g h &= \frac{m-m'}{t'-t} \pmod q \end{aligned} \tag{3.1}$$

Cele relatate mai sus subliniază faptul că partea care face un angajament pentru o valoare m, nu poate să ofere o deschidere pentru acel angajament diferită de deschiderea inițială (adică să furnizeze o altă pereche de valori $\pi' = (m', \alpha')$), astfel încât să se dezvăluie valoarea $m' \neq m$, decât dacă se cunoaște $\log_g h$.

Fiind cunoscute două angajamente $C = g^m h^t$ și $C' = g^m h^{t'}$, cu $t \neq t'$, pentru același mesaj inițial m, se poate demonstra că cele două ascund, într-adevăr, același mesaj fără a dezvălui valoarea acestuia $\iff$ se cunoaște valoarea $d = t - t'$. Astfel, cel care primește cele două angajamente calculează $\frac{C}{C'} = \frac{g^m h^t}{g^m h^{t'}} = \frac{h^t}{h^{t'}} = h^{t-t'} = h^d$, iar cel care a realizat angajamentele, publică valoarea d. Se observă că această valoare $d = t - t'$ nu dezvăluie nicio informație legată de mesajul conținut de către C și C'.

Pentru utilizarea eficientă a acestei construcții, în loc de grupul prezentat în lucrarea inițială [14], se pot folosi diverse grupuri ale curbelor eliptice în calculele angajamentului și verificării acestuia, fără a pierde din proprietățile de securitate ale sistemului inițial [8]. Fie $E(\mathbb{F}_p)$ grupului unei curbe eliptice alese, iar două puncte $E, P \in E(\mathbb{F}_p)$, cu $E = kP, k < q$,

k fiind un scalar necunoscut. Un angajament pentru o valoare m se calculează astfel: $C = mP + tQ$, unde + reprezintă adunarea a două puncte de pe curba eliptică, iar mP reprezintă înmulțirea unui punct cu un scalar.

3.2 Sisteme pentru angajamente de timp - Boneh-Naor

Sistemele pentru angajamente de timp (*timed commitment schemes*) sunt o extindere a sistemelor standard de angajamente. Acestea au aceleași funcționalități de bază (crearea unui angajament și deschiderea acestuia) și, în plus, îi oferă participantului care primește angajamentul posibilitatea de a accesa valoarea conținută de acesta prin intermediul unei acțiuni de *deschidere forțată*. Acțiunea de deschidere forțată presupune realizarea unui efort computațional secvențial considerabil, depinzând de un parametru de timp T, fără intervenția participantului care a creat angajamentul.

Aceste sisteme rezolvă următoarea problemă: Bob dorește să primească un mesaj de la Alice, iar aceasta creează un angajament pentru acel mesaj. Ea execută pașii necesari creării unui angajament, apoi îl transmite lui Bob. Mai apoi, ea poate alege să nu publice valoarea de deschidere a acestuia. Astfel, Bob va putea afla mesajul ales inițial de Alice doar prin rezolvarea unei probleme dificile. De asemenea, poate Alice nu are intenția de a trișa, însă pierde valoarea de deschidere a angajamentului. Așadar, un sistem care permite recuperarea forțată în timp a mesajului este o soluție viabilă, nefiind necesară valoarea de deschidere.

Formal, un sistem de angajament în timp [3] (T, t, ϵ) pentru un mesaj $M \in \{0,1\}^n$ îi oferă lui Alice posibilitatea de a-i oferi lui Bob un angajament asupra mesajului M; mai apoi, Alice poate să îi demonstreze lui Bob că M este mesajul conținut de angajament. De asemenea, în timpul etapei de deschidere, Alice nu poate să îl convingă pe Bob că angajamentul conține un alt mesaj $M' \neq M$. Însă, dacă Alice refuză să ofere valoarea de deschidere sau o pierde, Bob poate să execute acțiunea de recuperare forțată, obținând mesajul M în cel mult timp T. Alice este asigurată că, dacă Bob are la dispoziție un număr polinomial de procesoare, acesta va reuși să extragă mesajul M din angajament în timp t cu o probabilitate de cel mult ϵ [3].

Astfel de sisteme trebuie să satisfacă următoarele proprietăți [3]:

1. **Recuperare verificabilă**: după etapa de creare a angajamentului, participantul care îl primește este convins că, în urma acțiunii de deschidere forțată, se va dezvălui mesajul inițial ales de partea care s-a angajat. Cu alte cuvinte, mesajul care ar fi obținut în urma deschiderii standard coincide cu mesajul obținut în urma deschiderii forțate.

2. **Recuperare demonstrabilă**: după acțiunea de deschidere forțată, cel care realizează această acțiune obține atât mesajul conținut, cât și o *demonstrație* a acestei valori; astfel, oricine are acces la angajament, poate verifica faptul că mesajul obținut forțat este chiar cel conținut de către angajament, fără a executa și acesta acțiunea de deschidere forțată.

3. **Recuperare imună la atacuri paralele**: un participant cu mai multe procesoare nu este mai rapid în a recupera forțat valoarea conținută de angajament, decât unul cu un singur procesor, adică paralelizarea efortului computațional necesar acțiunii de deschidere forțată oferă un avantaj de timp neglijabil. Cu alte cuvinte, orice algoritm care poate fi paralelizat $\mathcal{A}$ al cărui timp de execuție este cel mult t cu $t < T$, care rulează pe un număr polinomial de procesoare, va reuși să distingă mesajul inițial M de un mesaj aleator $R \in \{0,1\}^n$, având la dispoziție angajamentul, cu un avantaj ϵ. Acest lucru poate fi formalizat astfel [3]:

$$\left|\Pr(\mathcal{A}(\text{angajament}, M) = \text{“da”}) - \Pr(\mathcal{A}(\text{angajament}, R) = \text{“da”})\right| < \epsilon$$

La fel ca în cadrul sistemelor de angajament clasice, un sistem de angajament în timp este alcătuit din (*i*) **etapa de angajament** și (*ii*) **etapa de deschidere**, la care se adaugă și (*iii*) **etapa de deschidere forțată** [3].

3.2.1 Inițializarea sistemului

Fie λ un număr natural reprezentând un parametru de securitate, iar $|x|$ reprezintă numărul de biți din scrierea binară a numărului natural x. Alice generează aleator două numere prime p, q cu $|p| = |q| = \lambda$ și $p \equiv q \equiv 3 \pmod 4$, apoi calculează $N = pq$ pe care îl trimite lui Bob. Un alt parametru public este B, care va fi folosit în cadrul sistemului atât de către Alice, cât și de Bob. De asemenea, este nevoie să se stabilească parametrul de timp T care reprezintă *timpul* necesar execuției etapei de deschidere forțată a angajamentului. Acesta poate fi stabilit fie de Alice, fie de Bob, fie de amândoi, în funcție de cadrul în care este folosit sistemul. În cadrul acestui sistem, dacă $T = 2^k$ atunci vor fi necesare 2^k înmulțiri modulare pentru a finaliza faza de deschidere forțată [3].

3.2.2 Etapa de angajament

În cadrul acestei etape, pe lângă realizarea angajamentului, Alice trebuie să îl convingă pe Bob că etapa de deschidere forțată va dezvălui cu succes mesajul conținut de angajament. De asemenea, Alice trebuie să se asigure că, dacă Bob are la dispoziție un număr polinomial de procesoare, atunci acesta nu va reuși să câștige un avantaj neneglijabil în timp pentru etapa de deschidere forțată. Astfel, acest sistem se bazează pe o problemă

care pare a fi inerent secvențială: exponențierea modulară. Pentru a calcula $x^{2^y} \pmod N$ utilizând cel mai bun algoritm cunoscut pentru exponențiere modulară, sunt necesare y ridicări la pătrat secvențiale.

Pentru a îndeplini cerințele prezentate mai sus, dacă Alice dorește să realizeze un angajament care să conțină un mesaj $M \in \{0,1\}^n$, atunci ea și Bob trebuie să parcurgă următorii pași [3]:

1. Alice calculează $Q = \prod_{i=1}^{b} q_i^N$, unde $q_1, q_2, ..., q_b$ reprezintă numerele prime mai mici decât B, apoi alege aleator un element $h \in \mathbb{Z}_N$ și calculează $g = h^Q \pmod N$. Astfel, Bob este sigur de faptul că $ord(g)$ în $\mathbb{Z}_N^*$ nu este divizibil cu niciun număr prim mai mic decât B.

2. Persoana care dorește să facă angajamentul calculează $u = g^{2^{2^k}} \pmod N$, calculând întâi valoarea intermediară $a = 2^{2^k} \pmod{\phi(N)}$, apoi $u = g^a \pmod N$, unde $\phi(N)$ reprezintă indicatorul lui Euler, cu $\phi(N) = (p-1)(q-1)$ deoarece p, q sunt prime, iar $N = pq$. Se observă că doar Alice cunoaște această valoare, păstrând secrete numerele p, q.

3. În cadrul acestui pas se determină valoarea S a angajamentului. Se construiește întâi o cheie de criptare K, $K = K_1K_2K_3...K_n \in \{0,1\}^n$ cu valorile $K_i = lsb(g^{2^{2^{k-i}}} \bmod N), \forall i \in \{1,2,...,n\}$, cu faptul că $g^{2^{2^{k-i}}} \bmod N$ se calculează prin obținerea valorii intermediare $a = 2^{2^{k-i}} \pmod{\phi(N)}$ și apoi $g^a \pmod N$, iar $lsb(x)$ reprezintă cel mai nesemnificativ bit din scrierea binară a numărului x. Valoarea S se obține aplicând operația *xor* între mesajul inițial M și cheia K, adică $S_i = K_i \oplus M_i$, $S = S_1S_2S_3...S_n \in \{0,1\}^n$. Alice trimite $C = (h, g, u, S)$ către Bob.

În acest punct, Alice trebuie să îi dovedească lui Bob faptul că algoritmul pentru etapa deschidere forțată se va încheia cu succes, dezvăluind valoarea M, fără a executa efectiv acel algoritm pentru a verifica acest fapt. Această proprietate a algoritmului se demonstrează prin verificarea că u este construit în mod corect, adică $u = g^{2^{2^k}} \pmod N$.

Întâi, persoana care a făcut angajamentul construiește șirul de valori

$$W = (g^2, g^4, g^{16}, g^{256}, ..., g^{2^{2^i}}, g^{2^{2^k}}) \mod N$$

de lungime $k+1$, folosind metoda valorii intermediare prezentate mai sus. Fie notația $W = (w_0, w_1, w_2, ..., w_k)$. Se observă că ultima valoare din acest șir este exact u, valoare pe care Alice trebuie să o demonstreze că este calculată conform protocolului. Astfel, arătând că fiecare triplet este de forma $(g, w_{i-1}, w_i), \forall i \in \{1,2,...,k\}$ în care se verifică $w_{i-1} = g^x, w_i = g^{x^2}, \forall i \in \{1,2,...,k\}$, adică $w_{i-1}^2 = w_i \pmod N, \forall i \in \{1,2,...,k\}$. Aceste k demonstrații nu oferă nicio informație legată de mesajul inițial M sau despre

cheia K (eng. orig. *zero-knowledge proof*). Protocolul complet pentru a demonstra corectitudinea șirului W necesită următorii cinci pași (în următoarele calcule, $q = ord(g)$ în $\mathbb{Z}_N^*$) [3]:

1. Bob alege k valori aleatoare $c_1, c_2, ..., c_k \in \{0, 1, ..., R\}$, unde R este un parametru de securitate ales de Bob. Acesta realizează câte un angajament pentru fiecare dintre aceste valori folosind un sistem pentru angajamente standard, precum [14].

2. Alice alege k valori aleatoare $\alpha_1, \alpha_2, ..., \alpha_n \in \mathbb{Z}_q$ și calculează perechile $(z_i, \omega_i)_{i=1}^k$, $z_i = g^{\alpha_i}, \omega_i = w_{i-1}^{\alpha_i} \pmod N$ pe care le trimite către verificator (Bob).

3. Bob îi oferă lui Alice deschiderile angajamentelor făcute asupra valorilor din șirul $c_1, c_2, ..., c_k$, dezvăluindu-le.

4. Acum că Alice are acces la șirul de valori dezvăluit de Bob, aceasta calculează și îi trimite lui Bob valorile $y_i = c_i \cdot 2^{2^{i-1}} + \alpha_i \pmod q$.

5. În final, Bob verifică egalitățile $g^{y_i} \cdot w_{i-1}^{-c_i} = z_i \pmod N$, cu extinderea în Ecuația 3.2, și $w_{i-1}^{y_i} \cdot w_i^{-c_i} = \omega_i \pmod N$, cu extinderea în Ecuația 3.3, $i \in \{1, 2, \dots, k\}$.

$$\begin{aligned}
g^{y_i} \cdot w_{i-1}^{-c_i} &= z_i && \pmod N \\
\left(g^{c_i \cdot 2^{2^{i-1}} + \alpha_i}\right) \cdot \left(g^{2^{2^{i-1}}}\right)^{-c_i} &= g^{\alpha_i} && \pmod N \\
g^{c_i \cdot 2^{2^{i-1}} + \alpha_i} \cdot g^{-c_i \cdot 2^{2^{i-1}}} &= g^{\alpha_i} && \pmod N \\
g^{c_i \cdot 2^{2^{i-1}} + \alpha_i - c_i \cdot 2^{2^{i-1}}} &= g^{\alpha_i} && \pmod N \\
g^{\alpha_i} &= g^{\alpha_i} && \pmod N
\end{aligned} \tag{3.2}$$

$$\begin{aligned}
w_{i-1}^{y_i} \cdot w_i^{-c_i} &= \omega_i && \pmod N \\
(g^{2^{2^{i-1}}})^{y_i} \cdot (g^{2^{2^i}})^{-c_i} &= w_{i-1}^{\alpha_i} && \pmod N \\
(g^{2^{2^{i-1}}})^{c_i \cdot 2^{2^{i-1}} + \alpha_i} \cdot (g^{2^{2^i}})^{-c_i} &= (g^{2^{2^{i-1}}})^{\alpha_i} && \pmod N \\
g^{2^{2^{i-1}}(c_i \cdot 2^{2^{i-1}} + \alpha_i)} \cdot g^{2^{2^i}(-c_i)} &= g^{\alpha_i 2^{2^{i-1}}} && \pmod N \\
g^{c_i \cdot 2^{2^i} + \alpha_i \cdot 2^{2^{i-1}}} \cdot g^{2^{2^i}(-c_i)} &= g^{\alpha_i 2^{2^{i-1}}} && \pmod N \\
g^{\alpha_i 2^{2^{i-1}}} &= g^{\alpha_i 2^{2^{i-1}}} && \pmod N
\end{aligned} \tag{3.3}$$

Lemă 3.2.1 *[3] Fie q ordinul lui g în $\mathbb{Z}_N^*$ și d cel mai mic divizor prim al lui q, atunci, dacă șirul de numere W este construit incorect, persoana care realizează angajamentul va*

reușI să îl păcălească pe verificator că este, de fapt, construit corect, cu o probabilitate de cel mult $k \cdot (\frac{1}{min(d,R)} + O(\frac{1}{R}))$.

În cadrul primului pas din etapa de deschidere, g a fost construit astfel încât ordinul acestuia nu este divizibil cu un număr prim mai mic decât B, așadar, cel mai mic divizor prim al lui q este mai mare decât B. Așadar, cu fiecare repetiție a protocolului care verifică faptul că șirul W a fost construit corect, Alice îl poate păcăli pe Bob cu o probabilitate de cel mult $\frac{1}{B}$. Pentru un nivel de securitate de 2^{-140}, se poate alege $B = 128$, iar protocolul să se execute de 20 de ori [3].

3.2.3 Etapa de deschidere

1. Bob calculează și el valoarea $Q = \prod_{i=1}^{b} q_i^N$. El a primit de la Alice $C = (h, g, u, S)$, iar protocolul de verificare din etapa de deschidere a demonstrat corectitudinea construirii valorii u.

2. Alice trimite valoarea $v' = h^{2^{2^k-n}}$ către verificator. Acesta calculează $v = {v'}^Q \bmod N$ și verifică faptul că $v^{2^n} = u \bmod N$. Practic, Bob a calculat $(h^{2^{2^k-n}})^Q \bmod N = (h^Q)^{2^{2^k-n}} \bmod N = g^{2^{2^k-n}} \bmod N = v$, apoi, ridicând acest rezultat la puterea 2^n ar trebui să obțină valoarea $u = g^{2^{2^k}} \pmod N$.

3. Având acum la dispoziție valoarea $v = g^{2^{2^k-n}} \bmod N$, Bob poate să calculeze fiecare componentă din cheia $K = K_1K_2K_3...K_n$ astfel: $K_i = lsb(v^{2^{(n-i)}}) = lsb((g^{2^{2^k-n}})^{2^{(n-i)}}) = lsb(g^{2^{(n-i)} \cdot 2^{2^k-n}}) = lsb(g^{2^{2^k-i}}), \forall i \in \{1, 2, ..., n\}$. Mai apoi, pentru a obține mesajul pentru care Alice a făcut angajamentul, se calculează $M_i = S_i \oplus K_i, \forall i \in \{1, 2, ..., n\}$.

Lemă 3.2.2 *[3] Mesajul M este legat de angajamentul $C = (h, g, u, S)$, adică un verificator poate deschide angajamentul doar la valoarea M.*

Demonstrația acestei leme se bazează pe faptul că dacă ordinul lui v în $\mathbb{Z}_N^*$ este impar, atunci u are o rădăcină de ordin 2^n unică în subgrupul generat de g, acea rădăcină fiind $v_0 = g^{2^{2^k-n}} \bmod N$. Mai mult, valoarea v' pentru care ține egalitatea din pasul 1 al etapei de deschidere este și ea unică.

3.2.4 Etapa de deschidere forțată

Alice poate alege să nu trimită valoarea $v' = h^{2^{2^k-n}}$ către verificator. Astfel, Bob calculează el însuși $v = g^{2^{2^k-n}} \bmod N$, apoi urmează aceeași pași ca în etapa de deschidere standard în vederea obținerii cheii $K = K_1K_2K_3...K_n$. Se observă că pentru a calcula valoarea v' sunt necesare $(2^k - n)$ ridicări la pătrat mod$N$ [3].

În cadrul acestei etape trebuie să arătăm că un adversar care are acces la un număr polinomial de procesoare și la un algoritm paralelizat nu poate să obțină nicio informație despre mesajul $M \in \{0,1\}^n$ conținut de angajamentul $C = (h, g, u, S)$ în mai puțin timp decât timpul necesar pentru 2^k ridicări la pătrat $\bmod N$. Mai întâi, se observă faptul că o cheie $K = K_1K_2K_3...K_n$ este, de fapt, o secvență pseudo-aleatoare generată de un generator **BBS** (denumit după autorii lucrării [2], L.Blum, M.Blum și M.Shub) pentru care ultima parte a secvenței generate este valoarea u. Demonstrația confidențialității mesajului are la bază presupunerea generalizată $(\lambda, n', \delta, \epsilon)$ **BBS** [3]:

Fie $g \in \mathbb{Z}$, un număr natural $k > n'$ și șirul de numere $W_{g,k} = (g^2, g^4, g^{16}, g^{256}, ..., g^{2^{2^i}}, g^{2^{2^k}}) \bmod N$. Atunci, pentru orice număr întreg n cu $n' < k < \lambda$ și orice algoritm paralelizat, rulat pe un număr polinomial de procesoare, $\mathcal{A}$, al cărui timp de execuție este mai mic decât $\delta \cdot 2^k$ avem că [3]:

$$\left|\Pr(\mathcal{A}(N, g, k, W_{g,k}, g^{2^{2^{k+1}}}) = \text{“da”}) - \Pr(\mathcal{A}(N, g, k, W_{g,k}, r^2) = \text{“da”})\right| < \epsilon \qquad (3.4)$$

unde $N = p \cdot q$ cu $|p| = |q| = \lambda$, cu p și q numere prime și $p \equiv q \equiv 3 \pmod 4$. Din această presupunere reiese faptul că, dat fiind șirul de numere $W_{g,k}$, un element $g^{2^{2^{k+1}}}$ este greu de deosebit de un element aleator r^2 pentru un algoritm paralelizat, rulat pe un număr polinomial de procesoare, $\mathcal{A}$, al cărui timp de execuție este mai mic decât 2^k [3]. Următoarea teoremă arată faptul că un adversar care are acces la un număr polinomial de procesoare și la un algoritm de tip prezentat anterior $\mathcal{A}$ al cărui timp de execuție este mai mic decât 2^k nu poate să obțină nicio informație despre mesajul $M \in \{0,1\}^n$ în acel timp:

Teoremă 3.2.1 *[3] Dacă presupunerea generalizată $(\lambda, n', \delta, \epsilon)$ **BBS** este valabilă pentru construcția prezentată, pentru anumite valori $\delta, \epsilon > 0$, atunci pentru un $k > n'$, sistemul prezentat este un sistem pentru angajamente de timp (T, t, ϵ) cu $t = \lambda \cdot 2^k$ si $T = M(\lambda) \cdot 2^k$, unde $M(\lambda)$ este timpul necesar ridicării la pătrat $\bmod N$ al unui număr reprezentat binar pe λ biți.*

Capitolul 4

Protocoale pentru licitații securizate

4.1 Prezentarea protocolului

Licitațiile de tip Vickrey prezentate anterior sunt predispuse la anumite riscuri de securitate în mediul online[18]:

- Dacă partea terță autorizată în care toți participanții au încredere are acces la valorile din ofertele acestora în timpul procesului de licitație, aceasta poate introduce o ofertă secretă cu doar ϵ mai mică decât valoarea depusă de către câștigător, maximizând astfel profitul vânzătorului.

- Dacă gazda nu are acces la valori, ea poate face anumite oferte strategice, urmând ca mai apoi să se folosească doar de cea care aduce cel mai mare profit vânzătorului, refuzând să le folosească pe celelalte.

- Un participant care este complice cu gazda (diferită de vânzător) poate trimite mai multe oferte, ca mai apoi să se publice doar cele oferte care îi avantajează pe amândoi (fie ca participantul să câștige neapărat sau ca acesta să nu câștige deoarece se răzgândește, realizând că prețul final va fi cu mult mai mare decât se aștepta să plătească).

Aceste probleme pot fi rezolvate ușor folosind protocoale securizate care asigură onestitatea, bazate pe sisteme pentru angajamente de timp. Un protocol securizat care asigură onestitatea pentru licitații online nu permite vânzătorului sau gazdei licitației să facă sau să schimbe anumite oferte în funcție de celelalte [3].

D. Boneh și M.Naor propun un astfel de protocol, bazat pe sistemul de angajamente de timp definit de aceștia [3]:

1. Participanții creează un angajament de timp pentru oferta lor și îl trimit pe acesta gazdei. Aceasta verifică validitatea angajamentului și respinge sau acceptă participanți în funcție de rezultatul verificării.

2. Când toți participanții au trimis angajamentele sau timpul pentru licitație se încheie, atunci gazda postează pe un *avizier* angajamentele și realizatorii acestora.

3. Dacă un participant își găsește numele pe avizier și vede că a fost acceptat, atunci acesta poate iniția etapa de deschidere, pentru a-și face publică oferta.

4. Dacă un participant refuză să trimită deschiderea angajamentului său, atunci gazda licitației poate să îl deschidă forțat pentru a publica oferta conținută de angajament.

Se observă că, indiferent de modul în care se alege prețul final la care se vinde bunul, acest lucru se poate face foarte ușor deoarece ofertele sunt publice la finalul procesului. Totodată, determinarea câștigătorului se face într-un mod cât se poate de transparent. Chiar dacă anumiți participanți ai licitației aleg să nu își deschidă angajamentul deoarece rezultatul final nu este benefic pentru aceștia sau aceștia nu pot să trimită valoarea de deschidere din motive obiective (de exemplu, dacă aceștia pierd accesul la valoarea $\phi(N)$), atunci proprietatea că un angajament făcut poate fi deschis în urma unui efort computațional împiedică fraudele sau greșelile.

Un aspect important pentru un astfel de protocol reprezintă alegerea parametrului de timp $T = 2^k$ în funcție de cât se dorește să dureze o licitație. În [3], se menționează că se așteaptă ca valoarea lui k să fie în intervalul $[30, \ldots, 50]$ pentru ca operația de deschidere forțată să dureze câteva ore sau câteva zile, în funcție de caz, dar nu este prezentată o analiză exactă a acestui lucru. În cadrul unei lucrări de actualitate despre protocoale pentru licitații securizate folosind angajamente de timp [18] se face totuși o scurtă analiză legată de numărul de ridicări la pătrat modulare pe secundă: pentru $|N| = 2048$ de biți, folosind cele mai bune implementări cunoscute, FPGA-urile (eng. orig. *Field Programmable Gate Array*) pot atinge 2^{24} ridicări la pătrat modulare, iar ASIC-urile (eng. orig. *Application Specific Integrated Circuit*) 2^{28}.

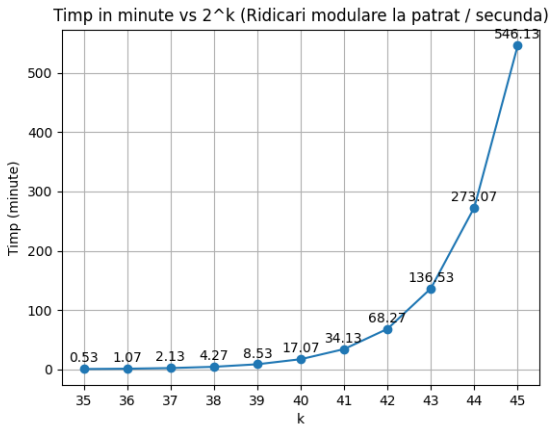

Figura 4.1: Analiza de timp a deschiderii forțate, pentru diferite valori ale lui $T = 2^k$, presupunând o putere de calcul de 2^{30} ridicări modulare la pătrat pe secundă.

Alegând o viteză de 2^{30} ridicări modulare la pătrat pe secundă, obținem următoarele estimări de timp pentru etapa de deschidere forțată, pentru $|N| = 2048$ de biți, prezentate în *Figura 4.1*.

4.2 Simularea protocolului

În cadrul simulării se creează patru containere de Docker care comunică între ele prin request-uri HTTPS, așa cum este prezentat și în *Figura 1.1*:

1. Un server care reprezintă gazda/vânzătorul licitației. Acesta primește ofertele pentru care s-au făcut angajamente de timp de la participanți.

2. Doi clienți care comunică doar cu serverul pentru a participa la licitație. Unul dintre cei doi clienți va iniția etapa de deschidere, iar celălalt va refuza să trimită valoarea de deschidere, astfel gazda va fi nevoită să deschidă forțat angajamentul.

3. Un container de Redis pe care serverul îl folosește drept bază de date pentru a stoca date necesare protocolului de licitație.

Pentru a realiza o simulare a protocolului prezentat, este nevoie de implementarea atât sistemului pentru angajamente Pedersen [14], cât și sistemului pentru angajamente de timp Boneh-Naor [3]. Acestea, precum și tot codul necesar simulării, se găsesc pe pagina mea de GitHub [1]. Pentru sistemul Pedersen, operațiile se fac pe curba eliptică ***SECP256k1*** din biblioteca *ecdsa-python* [20] pentru eficiență.

```
class PedersenCommitmentPublicParams:
    def __init__(self):
        curve = ecdsa.SECP256k1
        self.g = curve.generator
        self.N = curve.order
        random_power = secrets.randbelow(self.N - 1)
        while random_power <= 2:
            random_power = secrets.randbelow(self.N - 1)
        self.h = self.g * random_power
def pedersen_commit(message: int,pp: PedersenCommitmentPublicParams):
    alpha = secrets.randbelow(pp.N - 1)
    while alpha <= 2:
        alpha = secrets.randbelow(pp.N - 1)
    commitment = (pp.g * message)  + (pp.h * alpha)
    return commitment, alpha
def pedersen_open(commitment: int, message: int, opening: int, pp) :
    return commitment == pp.g * message + pp.h * opening
```

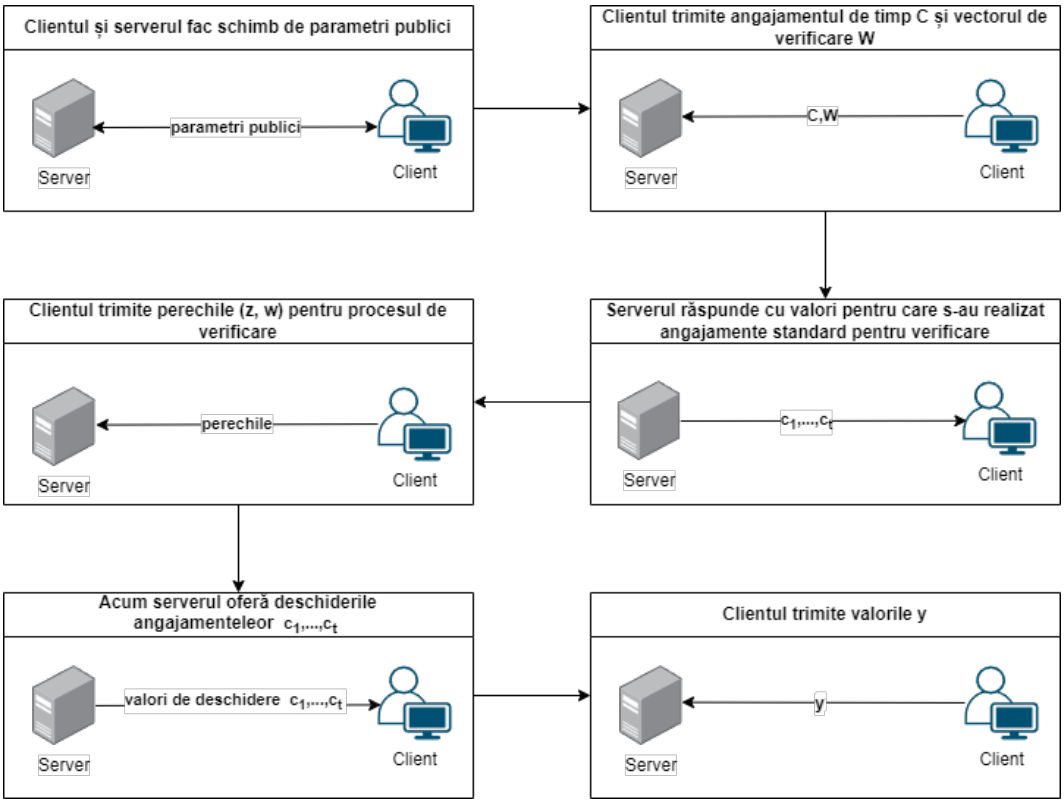

Figura 4.2: Ilustrarea etapelor de comunicare între server și client

Pentru început, gazda (serverul) stabilește parametrul de timp T pentru a asigura că toate angajamentele primite nu pot fi deschise înainte de a se termina licitația. De exemplu, în urma analizei parametrului de timp prezentată anterior se poate alege $k = 40$, iar licitația să aibă o limită de timp de 10 minute. Astfel, fiecare participant este convins că nimeni nu poate să plaseze oferte care să fie în funcție de oferta lui. Serverul primește request-uri HTTPS de la clienți pentru a se realiza comunicarea necesară urmării protocolului de licitație. Participanților care trimit angajamente li se atribuie anumite stări intermediare până la a fi acceptați sau respinși (în funcție de caz) în cadrul licitației. Astfel, serverul asigură onestitatea tuturor participanților prin verificarea veridicității angajamentelor primite, adică se verifică faptul că deschiderea forțată va dezvălui, într-adevăr, valoarea pentru care s-a făcut angajamentul. Etapele de comunicare între server și un client, până la momentul acceptării, sunt prezentate în *Figura 4.2*. După ce s-a terminat verificarea, unul dintre clienți așteaptă să se termine și timpul acordat licitației pentru a iniția etapa de deschidere, iar celălalt va refuza, serverul fiind nevoit să execute deschiderea forțată a angajamentului. Acest lucru se face manual într-o interfață grafică, așa cum este ilustrat în *Figura 4.3*, *Figura 4.4* și *Figura 4.5*.

Serverul salvează toate informațiile legate de clienți în baza de date Redis, iar găzduirea serverului web se face folosind biblioteca *Flask* [13] din Python. Clienții fac request-uri folosind biblioteca *requests* [15]. Pentru a securiza traficul web și a permite comunicarea prin HTTPS, am utilizat Ansible [4] pentru a automatiza procesul de generare a unui

certificat SSL self-signed. Serverul web a fost configurat să folosească acel certificat și cheia privată. Pentru clienți, la fiecare request HTTPS, am specificat certificatul CA (autoritatea de certificare) ca fiind certificatul serverului pentru a se putea realiza negocierea TLS.

Un exemplu de rută pentru un request HTTPS pe care gazda o servește este cel în care participantul inițiază etapa de deschidere:

```
@app.route('/open', methods=['POST'])
def open():
    try:
        req_json = request.get_json()
        client_id = req_json.get("client_id")
        v = req_json.get("v")
        client_data = get_client_data(client_id)
        if client_data is None:
            raise ValueError("Client ID not found.")
        if client_data["state"] != protocol_states.ACCEPTED:
            raise ValueError("Client must be first accepted to the
                auction in order to open his commitment.")
        verifier = client_data["verifier"]
        commitment = Commitment(client_data["commitment"]["g"],
            client_data["commitment"]["u"], client_data["commitment"][
            "S"])
        message = verifier.open(commitment, v)
        client_data["message"] = int(message, 2)
        save_client_data(client_id, client_data)
        return jsonify({"message": "Commitment opened to " + message,
            "client_id": client_id}), 200
    except Exception as e:
        return jsonify({"error": str(e)}), 400
```

Protocolul de licitație și simularea nu sunt influențate de numărul de clienți; totuși, din considerente legate de resurse, simularea are loc cu doar doi clienți.

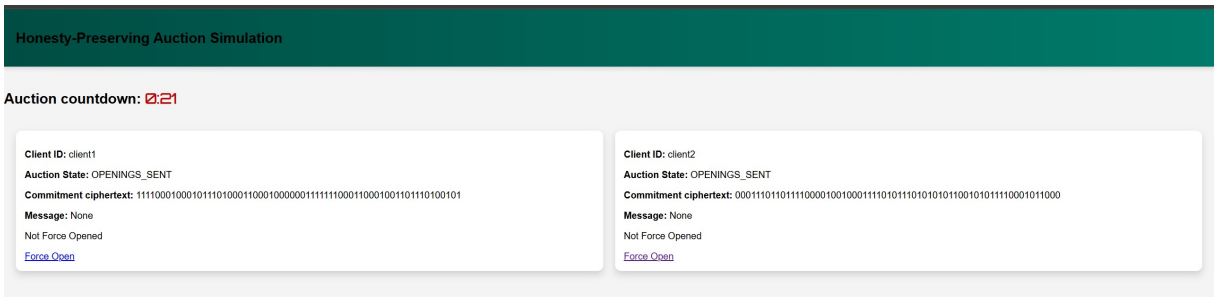

Figura 4.3: Stări intermediare în timpul protocolului

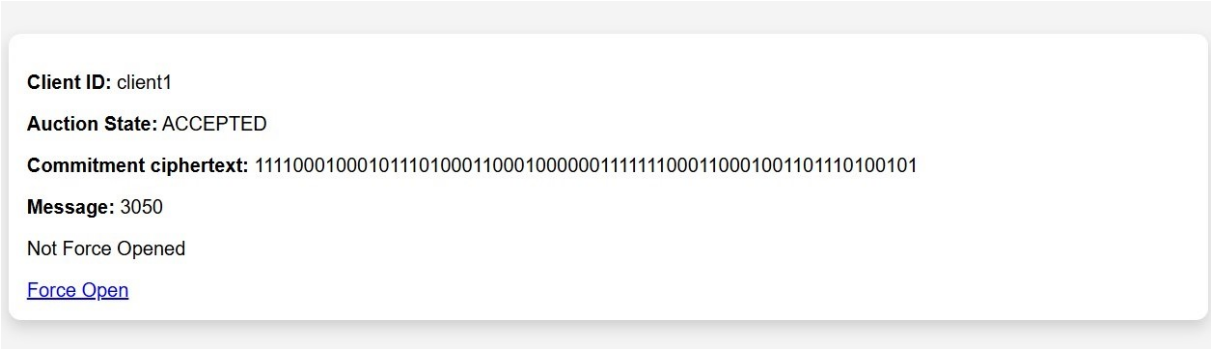

Figura 4.4: Clientul *client1* a inițiat faza de deschidere

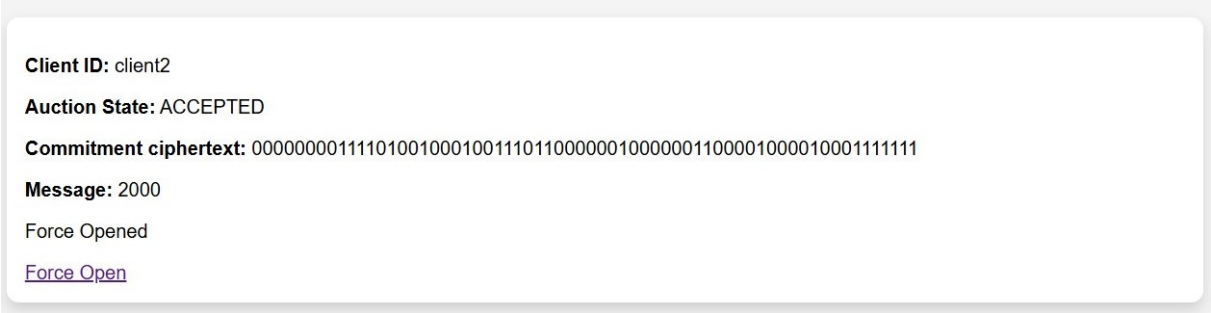

Figura 4.5: Pentru clientul *client2* s-a realizat deschidere forțată

4.3 Situația actuală a domeniului

O lucrare de actualitate care abordează tema protocoalelor securizate pentru licitații descentralizate este [18]. Aceasta se bazează tot pe sisteme pentru angajamente de timp, rezolvând o problemă pe care protocolul prezentat de D.Boneh și M.Naor o are, anume faptul că un astfel de sistem nu poate verifica faptul că un participant are destui bani cât să plătească suma din oferta făcută.

Rezolvarea folosește tehnologia **blockchain**, utilizând și alte tehnici precum demonstrații *zero-knowledge* de apartenență asupra unui interval (eng. orig. *zero-knowledge range-proofs*) și demonstrații de exponențiere (eng. orig. *proof of exponentiation*).

1. *Zero-knowledge range proofs* permit unui client să participe la mai multe licitații în același timp și asigură faptul că acel client are destui bani să plătească sumele din ofertele făcute. De asemenea, se pot utiliza pentru a impune anumite constrângeri unei licitații (de exemplu, ca suma din ofertă să fie minim 300).

2. Se folosesc *proofs of exponentiation* pentru a permite tuturor participanților să folosească același grup pentru calculele necesare angajamentelor, spre deosebire de sistemul prezentat de D.Boneh și M.Naor care necesită ca fiecare client să folosească un grup diferit, al cărui ordin să fie secret pentru fiecare. Astfel, se fac progrese din punct de vedere al eficienței.

Capitolul 5

Concluzii

Sistemele pentru angajamente de timp sunt o construcție importantă pentru domeniul criptografiei. Pe lângă utilizarea lor pentru protocoale pentru licitații securizate, acestea au și alte aplicații precum [3]: semnarea de contracte, *coin-flipping* și *zero-knowledge*. În cadrul protocoalelor pentru licitații acestea asigură un cadru onest din partea tuturor participanților și oferă confidențialitatea valorii inițiale a ofertei, oferind astfel încredere tuturor.

Lucrarea prezintă o implementare personală a unui protocol pentru licitații securizate și o simulare a acestuia, dezvăluind astfel capabilitățile sistemelor de angajamente și posibilitatea folosirii acestora în aplicații reale.

Un progres în domeniu se remarcă prin lucrări de actualitate care studiază astfel de sisteme, oferind plusuri de securitate și proprietăți care oferă o aplicabilitate mai extinsă a capabilităților. O direcție viitoare de dezvoltare este extinderea simulării la o aplicație reală care să permită utilizatorilor să participe la licitații găzduite într-un cadru onest și securizat. De asemenea, o altă posibilă direcție de dezvoltare pe viitor este folosirea unui sistem precum cel prezentat în [18] care ar aduce capabilități extinse pentru licitații.

Bibliografie

[1] Apostol Alin-Constantin, *Cod Licență Apostol Alin-Constantin*, https://github.com/Apostol-Alin/Licenta/tree/master/boneh, Accesat: 05.06.2025.

[2] Lenore Blum, Manuel Blum și Mike Shub, „A simple unpredictable pseudo-random number generator", în *SIAM Journal on computing* 15.2 (1986), pp. 364–383.

[3] Dan Boneh și Moni Naor, „Timed commitments", în *Annual international cryptology conference*, Springer, 2000, pp. 236–254.

[4] Ansible project contributors, *Ansible documentation*, https://docs.ansible.com/, Accesat: 09.06.2025.

[5] Gilad Edelman, *Google's Alleged Scheme to Corner the Online Ad Market*, https://www.wired.com/story/google-antitrust-ad-market-lawsuit/, Accesat: 22.04.2025.

[6] Diffie-Hellman Key Exchange, „Diffie-Hellman Key Exchange", în *Diffie% E2* 80 (1976).

[7] Python Software Foundation, *Python 3.13.3 documentation*, https://docs.python.org/3/, Accesat: 22.04.2025.

[8] Christian Franck și Johann Großschädl, „Efficient Implementation of Pedersen Commitments Using Twisted Edwards Curves", în Sept. 2017, pp. 1–17, ISBN: 978-3-319-67806-1, DOI: 10.1007/978-3-319-67807-8_1.

[9] Redis Inc, *Redis documentation*, https://redis-py.readthedocs.io/en/stable/, Accesat: 22.04.2025.

[10] Docker Inc., *Docker documentation*, https://docs.docker.com/, Accesat: 22.04.2025.

[11] Paul R Milgrom și Robert J Weber, „A theory of auctions and competitive bidding", în *Econometrica: Journal of the Econometric Society* (1982), pp. 1089–1122.

[12] NIST, *National Institute of Standards and Technolohy (NIST) – Computer Security Resource Center (CSRC), Glossary.* https://csrc.nist.gov/glossary/term/cryptography, Accesat: 22.04.2025.

[13] Pallets, *Flask documentation*, https://flask.palletsprojects.com/en/stable/, Accesat: 22.04.2025.

[14] Torben Pryds Pedersen, „Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing”, în *Advances in Cryptology — CRYPTO '91*, ed. de Joan Feigenbaum, Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 129–140, ISBN: 978-3-540-46766-3.

[15] MMXVIX. A Kenneth Reitz Project, *Requests documentation*, https://requests.readthedocs.io/en/latest/, Accesat: 22.04.2025.

[16] Office of Public Affairs, *Department of Justice Prevails in Landmark Antitrust Case Against Google*, https://www.justice.gov/opa/pr/department-justice-prevails-landmark-antitrust-case-against-google, Accesat: 22.04.2025.

[17] SymPy Development Team, *SymPy documentation*, https://docs.sympy.org/latest/index.html, Accesat: 22.04.2025.

[18] Nirvan Tyagi, Arasu Arun, Cody Freitag, Riad Wahby, Joseph Bonneau și David Mazières, „Riggs: Decentralized sealed-bid auctions”, în *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1227–1241.

[19] William Vickrey, „Counterspeculation, auctions, and competitive sealed tenders”, în *The Journal of finance* 16.1 (1961), pp. 8–37.

[20] Brian Warner și Hubert Kario, *Python-ECDSA documentation*, https://ecdsa.readthedocs.io/en/latest/, Accesat: 22.04.2025.