UNIVERSITATEA POLITEHNICA BUCUREȘTI
ȘCOALA DOCTORALĂ DE AUTOMATICĂ ȘI CALCULATOARE

Sumar Teză de doctorat

Securizarea infrastructurii IoT prin soluții blockchain

Autor: Drd. CSIII ing. Florin Răstoceanu

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

BUCUREȘTI

2023

Cuprins

Abstract

Internetul lucrurilor (IoT) se află în continuă dezvoltare, influențându-ne din ce în ce mai mult calitatea vieții printr-o multitudine de aplicații utile și ușor de utilizat, dar în același timp, expunându-ne la amenințări care ne-ar putea periclita securitatea datelor personale. Specificitatea arhitecturilor IoT, ce presupune interconectarea elementelor din cadrul rețelelor de calculatoare clasice cu ființe vii, impune asigurarea unui grad de securitate sporit. Acest lucru este îngreunat de anumite aspecte specifice arhitecturilor IoT, cum ar fi eterogenitatea dispozitivelor IoT sau resursele limitate avute la dispoziție. Nevoia de securitate este cu atât mai acută cu cât atacurile cibernetice care vizează dispozitive IoT au crescut în mod exponențial în ultima perioadă.

În această teză propun o serie de soluții menite să îmbunătească securitatea în infrastructurile IoT. Serviciile de securitate de bază precum confidențialiatea, integritatea, autentificarea și non-repudierea se pot implementa folosind protocoale și mecanisme criptografice. Acestea pot asigura gradul de securitate dorit doar dacă utilizează chei criptografice și parametri de intrare aleatorii. Având în vedere aceste aspecte, dar și specificitatea mediului IoT, abordez problema securității pe mai multe paliere.

În primul rând, identific modalități de integrare a tehnologiei blockchain cu o arhitectură IoT pentru a asigura suport pentru implementarea serviciilor de securitate. Utilizarea unor noduri IoT care să asigure și funcționalități specifice blockchain a fost analizată din punctul de vedere al resurselor și costurilor. În acest sens propun utilizarea nodurilor IoT din cadrul unei arhitecturi de tip fog computing ca noduri blockchain cu funcționalități adaptate resurselor avute la dispoziție și implementarea nodurilor IoT folosind două tipuri de arhitecturi FPGA. Registrul blockchain este utilizat ca sursă de încredere în cadrul implementării unui protocol simplu și sigur de negociere chei de sesiune. Soluția asigură securitate sporită prin folosirea de primitive sigure din punct de vedere criptografic. Datorită simplității soluției și selectării unor funcții criptografice adecvate se oferă un consum de putere optimizat pentru dispozitive IoT care dispun de resurse limitate.

Eficacitatea algoritmilor criptografici este în strânsă legătură cu cheile criptografice utilizate, care trebuie să fie aleatoare și să nu poată fi deduse de potențiali atacatori. Acestea pot fi generate doar utilizând generatoare aleatoare evaluate corespunzător, care să prezinte caracteristici de securitate și eficiență sporite. Având în vedere aceste aspecte, propun o variantă de generator de numere aleatoare, care folosește un algoritm de criptare lightweight și care îndeplinește proprietăți de securitate ridicate prin reinițializarea intrărilor cu entropie proaspătă la fiecare apelare.

Asigurarea unui grad dorit de impredictibilitate pentru intrările generatoarelor de numere aleatoare se poate obține folosind surse capabile să ofere un nivel de entropie corespunzător. În acest sens, sursa de entropie trebuie să fie stabilă și rezistentă la atacuri iar pentru utilizarea în medii IoT trebuie să fie și eficientă. Pentru a acoperi aceste caracteristici soluția propusă în teză folosește aleatorismul generat de senzori de mișcare. Astfel, eficiența este asigurată prin folosirea unor resurse existente în platformele IoT. Utilizând o metodologie originală și exaustivă de analiză a sursei de entropie din punctul de vedere al sursei de zgomot, stabilității și rezistenței la atacuri, am validat sursa de entropie pentru utilizare în aplicațiile IoT care folosesc senzori de mișcare.

Soluțiile propuse în teză pot asigura protejarea datelor vehiculate în medii IoT prin utilizarea registrului blockchain ca ancoră de încredere în cadrul unui protocol de stabilire chei și a unor mecanisme sigure de generare de numere aleatoare și entropie. Eficiența consumului de putere, dovedită prin implementări pe platforme specifice, califică aceste soluții ca fiind potrivite pentru utilizarea în aplicații IoT.

1. Introducere

Internetul lucrurilor (Internet of Things - IoT) se răspândește cu rapiditate, fiind prezent din ce în ce mai mult în viețile noastre. Un *lucru* poate fi un dispozitiv inteligent (ceas inteligent, imprimantă, frigider, mașină de spălat, automobil, dronă, casă inteligentă, încuietoare inteligentă, etc), un implant care monitorizează și reglează bătăile inimii sau nivelul de zahăr din sânge ale unei persoane sau un cip inteligent implantat unui animal într-o fermă. Un *lucru* poate fi considerat ca făcând parte din IoT dacă este conectat la o rețea și are capabilitatea de a face schimb de date cu alte componente din sistem [1]. Folosind senzori și actuatori, prin intermediul infrastructurii IoT, se realizează o legătură între Internet, privit ca o rețea globală de calculatoare, și alte aparate cu adrese computerizate, mediul natural, reprezentat de oameni, animale sau elemente din natură [2].

Infrastructura IoT presupune interconectarea rețelelor de calculatoare cu ființe vii sau cu elemente din natură. Astfel, riscurile existente în mediul cibernetic se tranferă și către acestea din urmă, aducând prejudicii mult mai grave și mult mai greu de contracarat. Asigurarea securității în astfel de sisteme este cu atât mai importantă cu cât adopția acestei tehnologii în viața noastră este realizată într-un ritm din ce în ce mai alert. Conform *www.statista.com* [3], în anul 2020 erau active aproximativ 9,7 miliarde de dispozitive IoT și se estimează că numărul lor se va tripla în zece ani, ajungând până la 29,4 miliarde. Pe de altă parte, atacurile cibernetice care vizează dispozitive IoT au crescut în mod alarmant în ultima perioadă. De exemplu, Symantec a raportat o creștere cu 600% a atacurilor din 2016 în 2017 [4], iar în prima jumatate a anului 2021, Kaspersky a raportat 1,5 miliarde de atacuri desfășurate împotriva dispozitivelor IoT.

1.1. Motivația tezei

Asigurarea securității într-un mediu atât de divers și de complex se lovește de multe aspecte specifice dispozitivelor IoT. În primul rând acest sistem este puternic eterogen. În prezent pe piață există o multitudine de dispozitive care diferă prin sistemele de operare, interfețele de rețea, protocoalele utilizate, mecanismele și funcțiile de securitate implementate. Pentru a rezolva aceste probleme trebuie identificate tehnologii noi care se mulează pe arhitecturile de securitate IoT. Așa cum este specificat și în raportul National Institute of Standards and Tehnology (NIST) , referitor la standardizarea securității cibernetice pentru IoT [1], tehnologia blockchain (BC) are un potențial semnificativ în acest domeniu. Asigurarea securității în mod descentralizat oferă anumite avantaje mediului IoT în comparație cu soluțiile clasice ce se bazează pe infrastructuri de chei publice (Public Key Infrastructure - PKI). Folosind tehnologia blockchain anumite dezavantaje pot se pot transforma în avantaje. Astfel, numărul mare de dispozitive IoT poate fi benefic pentru a asigura o descentralizare cât mai bună și a spori gradul de încredere în soluția implementată, dar poate oferi și o disponibilitate ridicată, asigurând un număr de suficient de entități care să valideze tranzacțiile. Caracteristica de imutabilitate poate oferi posibilitatea de a stoca date care nu pot fi modificate, fapt ce poate fi foarte util pentru funcțiile de auditare a evenimentelor. Chiar și caracterul eterogen al infrastructurilor IoT poate

fi asimilat, deoarece tehnologia blockchain are nevoie doar de o adresă și de capacitatea de comunicare în rețele peer –to- peer.

Pe de altă parte, cu puține excepții, dezvoltatorii sunt orientați spre asigurarea interconectării dispozitivelor și mult mai puțin spre asigurarea securității acestora. Acest lucru este greu de realizat dacă luăm în considerare faptul că multe dintre acestea dispun de capacitate de procesare și stocare redusă sau nu au la dispoziție o sursă de alimentare permanentă, funcționând pe baterie. În acest caz, implementarea criptografiei clasice este aproape imposibil de realizat, fiind necesară identificarea unor soluții pentru algoritmi, protocoale și mecanisme criptografice adaptate unor dispozitive cu resurse limitate. Criptografia se folosește pentru a asigura servicii de securitate de bază cum ar fi: confidențialitatea, integritatea, autentificarea și non-repudierea. Pentru a asigura robustețea și rezistența acestora la diferite tipuri de atacuri trebuie avută în vedere întreaga infrastructură de securitate. În Figura 1.1 sunt prezentate toate elementele care o compun.

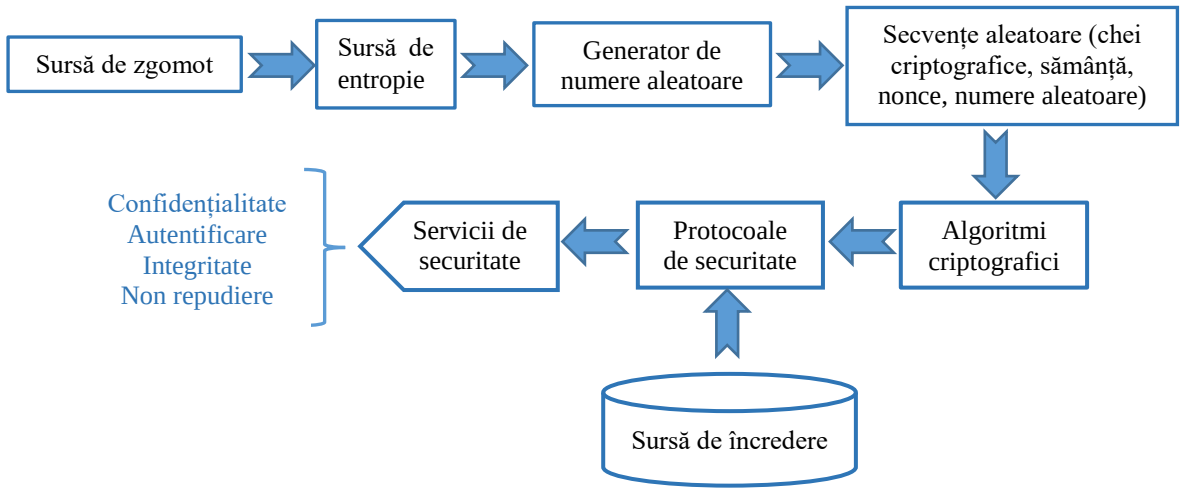

Figura 1.1 Flux de implementare a serviciilor de securitate

1.2. Obiectivele tezei

Scopul tezei este de a identifica soluții pentru asigurarea securității în infrastructuri IoT. Acest obiectiv poate fi realizat pe mai multe niveluri, având în vedere fluxul de implementare al serviciilor de securitate așa cum este prezentat în Figura 1.1 și constrângerile specifice mediului IoT.

Primul obiectiv este de a identifica soluții de integrare a tehnologiei blockchain în infrastructuri IoT, cu scopul de a asigura metode sigure și eficiente pentru a implementa servicii de autentificare mutuală și confidențialitate a comunicațiilor între diferite dispozitive IoT. Acest obiectiv se poate realiza prin:

- Identificarea modalităților optime prin care tehnologia blockchain poate oferi suport unei infrastructuri IoT pentru a aduce beneficii superioare soluțiilor clasice care folosesc PKI;
- Propunerea unei arhitecturi comune IoT – BC și optimizarea acesteia din punct de vedere al resurselor și costurilor;

- Identificarea unor soluții arhitecturale optime pentru platforme hardware, utilizate pentru implementarea unor noduri IoT dar care să prezinte și funcționalități specifice nodurilor BC.

Al doilea obiectiv este de a identifica o soluție mai simplă, dar în același timp sigură, pentru un protocol utilizat pentru stabilirea cheilor de criptare și autentificare, necesare securizării comunicațiilor între nodurile IoT. Acest obiectiv se poate realiza prin:

- Asigurarea compatibilității cu tehnologia blockchain;
- Integrarea optimă cu dispozitivele IoT cu resurse limitate;
- Optimizarea consumului de putere utilizând mecanisme și funcții criptografice adecvate;
- Asigurarea unui nivel de securitate care să nu permită compromiterea datelor vehiculate.

Al treilea obiectiv este de a identifica un generator de numere aleatoare care să asigure cel mai înalt grad de securitate, dar care să poată fi implementat pe dispozitive IoT cu resurse limitate. Acest obiectiv se poate realiza prin:

- Identificarea unei soluții care să asigure un grad de securitate sporit, având în vedere aleatorismul datelor generate și tăria criptografică a componentei deterministe a generatorului;
- Optimizarea eficienței generatorului de numere aleatoare din punct de vedere al resurselor consumate raportate la viteză și volumul de date generate.

Al patrulea obiectiv este de a identifica o sursă de entropie care să poată fi implementată cu resurse minime pe dispozitive IoT, dar care să asigure un nivel de entropie suficient de bun pentru a fi folosită în context criptografic. Acest obiectiv se poate realiza prin:

- Identificarea unei surse de entropie care să utilizeze cât mai puține resurse, eventual să utilizeze din resursele deja existente pe nodurile IoT;
- Estimarea nivelului de entropie generat folosind metodologii standardizate și de încredere;
- Optimizarea eficienței sursei de entropie prin parametrizarea optimă în diferite cazuri de funcționare;
- Aplicarea unei metodologii de testare și evaluare a sursei de entropie pentru a analiza comportamentul acesteia în diferite cazuri de utilizare, pe termen lung, și rezistența la diferite tipuri de atacuri.

2. Aspecte privind securitatea infrastructurilor IoT

În cadrul acestui capitol am prezentat aspecte esențiale referitoare la securitatea infrastructurilor IoT. Am tratat problematica din punctul de vedere al stadiului actual, prezentând și noțiuni teoretice care definesc conceptele elaborate în capitolele următoare din teză. Astfel, sunt definite domeniile de securitate cibernetică cu aplicare în IoT, principalele abordări arhitecturale în IoT dar și aspecte esențiale referitoare la asigurarea serviciilor de securitate, cu accent pe managementul cheilor criptografice în general și în cadrul protocoalelor utilizate în IoT, modalități de generare a numerelor aleatoare și algoritmi criptografici

lightweight. În final este abordată tehnologia blockchain și modalitățile de aplicare a acesteia pentru securizarea infrastructurilor IoT

2.1. Domenii de securitate cibernetică în IoT

Pentru a asigura securitatea într-un ecosistem IoT trebuie luate în considerare mai multe aspecte conform raportului NIST, referitor la standardizarea mediului de securitate cibernetică IoT [1]. Printre cele mai importante sunt următoarele:

- tehnicile criptografice implementate pentru asigurarea protecției datelor sensibile stocate sau transmise. Cea mai mare provocare, în acest caz, este dată de resursele limitate specifice multora dintre componentele aplicațiilor IoT;
- evaluare de securitate care are drept scop asigurarea implementării mecanismelor de securitate în sistemul sau produsul IT, efectuarea unor teste de securitate care să valideze un anumit nivel de securitate, aplicarea unor metrici universale de măsurare a tăriei mecanismelor și a funcțiilor criptografice implementate;
- protecția fizică care pot proteja dispozitivele IoT împotriva unor atacuri pasive sau intruzive realziate pentru a extrage cheile criptografice sau datele sensibile. Pentru a preveni astfel de atacuri se pot aplica filtre pe alimentare sau se pot construi carcase care să nu permită emisia de radiații în exterior. În [5] am realizat un studiu care simulează și modelează câmpul electromagnetic al unei carcase de nod de senzori
- Securitatea componentelor hardware/software prin care se asigură că nu au vulnerabilități cunoscute;
- Managementul identităților și accesului prin care se asigură un acces discreționar la date diferitelor entități reprezentate persoane, organizații, dispozitive hardware, aplicații software;
- Securitatea rețelei prin care se asigură managementul, operarea și utilizarea în siguranță a datelor;
- Managementul riscului privind dezvoltarea și livrarea produselor, care privește aspecte referitoare la modalitățile prin se care se asigură livrarea unor produse conforme cu specificațiile

2.2. Arhitecturi de securitate specifice infrastructurilor IoT

Majoritatea lucrărilor științifice abordează securitatea în strânsă legătură cu arhitectura sistemului. Arhitecturile prezentate în literatură sunt compuse dintr-un număr de 3, 4 sau 5 niveluri.

Arhitecturile mai simple sunt prezentate pe trei niveluri de bază: percepție, rețea și aplicație. Nivelul percepție este prezent în orice arhitectură și reprezintă legătura cu mediul. Acesta este alcătuit din senzori și actuatori înglobați în nodul de senzori, care are capacitatea de a transmite

datele acumulate următorului nivel de transport sau rețea. Nivelurile transport/rețea sunt cele care asigură conectivitatea și transmiterea mesajelor către celelalte niveluri. Pe al treilea nivel rulează diferite aplicații care agreghează datele acumulate de la senzori și le dau un scop clar și precis. La acest nivel sunt realizate servicii de mentenanță a aplicațiilor, de acces control și actualizări de securitate software.

Pentru a putea integra mai bine caracterul eterogen și complex al sistemelor IoT au fost introduse niveluri suplimentare. Astfel, după nivelul de transport, a apărut un nivel suplimentar care are rol de procesare intermediară a datelor înaine de a fi trimise la nivelul aplicație. Acesta poate fi identificat în literatură sub diferite denumiri: middleware (nivel intermediar), procesare sau servicii. Acest nivel asigură interoperabilitatea și scalabilitatea necesare pentru a oferi servicii utilizatorilor făcând abstracție de componenta hardware. Tot pe acest nivel se asigură managementul serviciilor și accesul la baze de date.

O a treia abordare arhitecturală a sistemelor IoT este cea cu cinci niveluri. Într-o astfel de arhitectură este introdus un nivel suplimentar deasupra nivelului de aplicație. În cele mai multe lucrări poate fi identificat cu numele de nivelul business. Acest nivel este responsabil cu managementul sistemul IoT în ansamblu, punând la dispoziție modele de bussines, grafice, tabele de date structurate pe baza informațiilor provenite de la nivelul aplicație. Într-o altă abordare pe cinci niveluri ultimul nivel este interfața cu utilizatorul.

2.3. Managementul cheilor criptografice

Analiza securității unui sistem criptografic pleacă de la premisa că algoritmii criptografici sunt cunoscuți și că tăria acestuia rezidă în capacitatea de a proteja cheile criptografice folosite. În acest sens utilizarea și protecția cheilor criptografice pe întregul ciclu de viață al acestora este la fel de importantă ca protecția datelor sensibile. Ciclul de viață al cheilor criptografice, prezentat în Figura 2.1, cuprinde, în funcție de utilizarea cheilor, următoarele etape: generare, stocare, transport, import, export, utilizare și distrugere sau zeroizare. Generarea trebuie realizată doar folosind generatoare care oferă un caracater aleator cheilor dar și care, în funcție de aplicație, oferă și alte proprietăți, așa cum au fost menționate în capitolul anterior. Pe timpul stocării, cheile criptografice trebuie protejate prin criptare sau folosind diferite mecanisme hardware specifice care nu permit accesul la acestea sau îl permit doar după autentificare. Pe timpul transportului în medii neprotejate trebuie asigurată criptarea, integritatea și autentificarea cheilor criptografice. Anumite module criptografice nu permit importul sau exportul cheilor decât în format criptat și autentificat. După utilizarea cheilor criptografice, acestea trebuie distruse prin zeroizare, care se poate realiza prin suprascriere.

Modalitățile de implementare a unor sisteme de management al cheilor sunt diferite de la caz la caz. Cheile simetrice trebuie să se afle în posesia corespondenților. Modalitatea de distribuție a acestora se poate realiza electronic sau manual folosind alte metode de protecție pe timpul transportului. Distribuția prin metode electronice necesită asigurarea unor servicii de confidențialitate, integritate și autentificare, care se pot obține folosind alte chei criptografice, care sunt de cele mai multe ori chei asimetrice. Acestea trebuie de asemenea distribuite. În acest caz, distribuția se realizează folosind o infrastructură de chei publice (în engleză Public

Key Infrastructure – PKI). Un sistem PKI este o structură centralizată de management al certificatelor digitale.

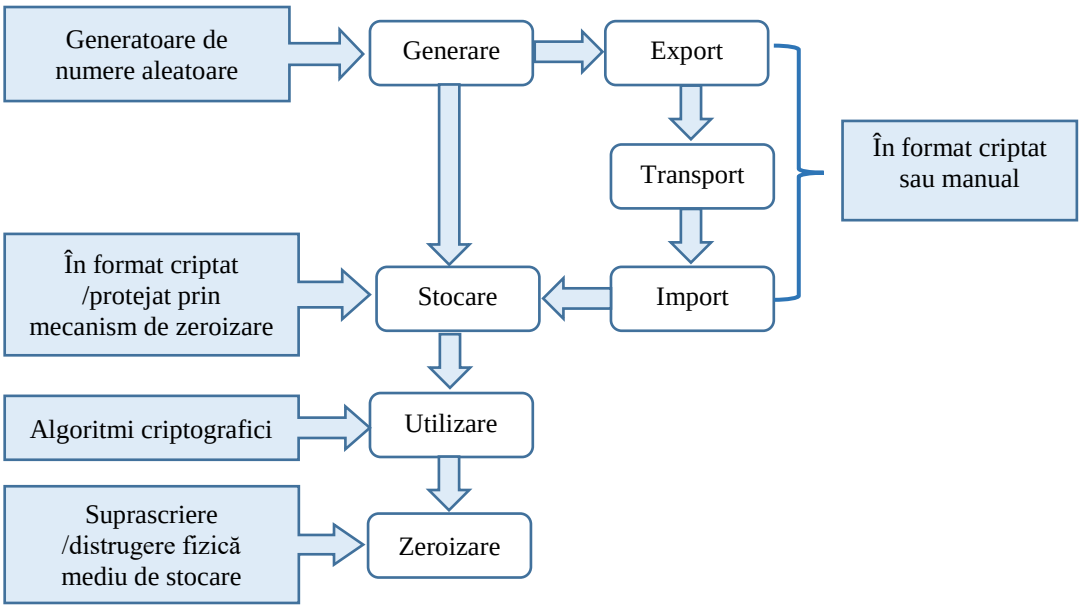

Figura 2.1 Ciclul de viață al cheilor criptografice

2.4. Metode de generare a cheilor criptografice

2.4.1. Generatoare de numere aleatoare

Conform NIST [6], un generator de numere aleatoare este un dispozitiv sau un algoritm capabil să genereze secvențe aleatoare de biți care au proprietățile de a fi independenți din punct de vedere statistic și de a fi distribuiți uniform. Generatoarele de numere aleatoare pot fi de două feluri: deterministe și non-deterministe. Generatoarele de numere aleatoare deterministe (în engleză Deterministic Random Number Generator - DRNG) sunt construite pe baza unui algoritm care folosește o valoare aleatoare inițială secretă, numită și sămânță, pentru a genera secvențe mai lungi de numere aleatoare. Acestea se mai numesc și generatoare de numere pseudoaleatoare. Generatoarele de numere aleatoare non-deterministe (în engleză True Random Number Generator – TRNG) folosesc surse de entropie, bazate pe surse de zgomot obținut din fenomene fizice aleatoare sau evenimente cu caracter aleator.

Dacă sursele non-deterministe pot genera numere aleatoare cu entropie maximă, ele au o viteză de generare mică în raport cu nevoile celor mai multe aplicații criptografice. Din acest motiv se folosesc generatoare deterministe care pot genera cu viteze mult mai mari și care folosesc intrări provenite de la surse de zgomot care pot fi surse de entropie sau generatoare non-deterministe.

Sursele de entropie sunt compuse dintr-o sursă de zgomot, o funcție condițională, care poate fi opțională, și o suită de teste de sănătate. Sursa de zgomot este elementul care generează aleatorismul sursei de entropie. Aceasta conține elementele care dau caracterul non-determinist datelor generate de sursa de entropie. Sursele de zgomot pot fi clasificate în două tipuri: software și hardware. Cele software își extrag aleatorismul din caracterul aleator al diferitelor procese și evenimente specifice sistemelor de operare. Aceste tipuri de surse de zgomot

necesită sisteme de operare pe care rulează multe procese sau unde intervenția operatorilor este frecventă. Deoarece activitatea operatorilor pe dispozitivele IoT este redusă, procesele sunt limitate datorită consumului de putere sau nu există sisteme de operare. Aceste soluții nu sunt pretabile pentru utilizarea în IoT. În mod uzual sursele de zgomot sunt bazate pe fenomene fizice a căror desfășurare prezintă un caracter aleator. Printre soluțiile existente se pot menționa cele care folosesc diode, circuite FPGA sau aleatorismul unor senzori folosiți în IoT [7][8].

2.4.2. Generatoare de numere aleatoare în contextul IoT

Problema generării numerelor aleatoare pe dispozitive cu resurse limitate a mai fost abordată în mai multe studii. Soluțiile prezentate în aceste studii folosesc date achiziționate de la diferite tipuri de senzori sau de la surse de zgomot aleatoare, existente pe platformele IoT în cauză. Din analiza acestor soluții, ținând cont de proprietățile pe care trebuie să le îndeplinească generatoarele de numere aleatoare folosite în aplicații criptografice, se pot concluziona mai multe aspecte:

- Metodologia de estimare a entropiei nu folosește un număr suficient de estimatori sau în unele cazuri nu este menționată. Folosirea unui număr mai mic de estimatori poate influența rezutatele, deoarece valoarea finală a entropiei estimate este dată de estimatorul care a obținut cea mai mică valoare;
- Cantitatea de date analizate nu este suficientă în majoritatea cazurilor. Pentru a obține rezultate statistice corespunzătoare este necesară o cantitate mare de date pentru analiză. Metodologia NIST impune un număr minim de 1.000.000 de secvențe;
- Niciuna dintre lucrările analizate nu a prezentat o analiză a valorii entropiei în condiții diferite de setare a parametrilor senzorilor. Parametrii precum lățimea de bandă sau intervalul de măsurare pot influența categoric valoarea entropiei obținută de la senzori.
- Un singur studiu a analizat din punct de vedere teoretic rezistența sursei de entropie la atacuri de tip canal alăturat. Această analiză este foarte importantă, deoarece poate oferi informații despre posibilitatea ca un atacator să poată aproxima datele colectate de senzori și în acest fel datele de ieșire ale sursei de entropie;
- Un singur studiu a analizat influența frecvenței de eșantionare și a temperaturii asupra valorii entropiei;

2.5. Algoritmi criptografici lightweight

Infrastructura IoT conține o multitudine de dispozitive care au resurse limitate, cum ar fi dispozitivele RFID sau diferite tipuri de senzori. Aceste dispozitive alocă foarte puține resurse pentru asigurarea securității datelor. Astfel, s-a creat o nevoie de dezvoltare a unor algoritmi criptografici care să păstreze un nivel suficient de securitate dar care să nu folosească prea mult resurse. În concluzie o implementare este un compromis între securitate, performanțe și costuri. Un algoritm cu performanțe bune și costuri mici va fi expus la atacuri de tip side-channel. Dacă se implementează măsuri de prevenire a acestor tipuri de atacuri, cresc costurile și scade

performanța. Rolul algoritmilor de tip lightweight este de a găsi soluția optimă pentru a atinge toate aceste obiective într-un mod satisfăcător. În funcție de modalitatea de implementare, software sau hardware, eficiența poate fi evaluată diferit. Pentru implementările hardware contează consumul de memorie și dimensiunea implementării, care se exprimă în numărul de porți utilizate. Acesta trebuie să fie într-un număr cât mai mic. Alți parametri importanți sunt:

- viteza de procesare, exprimată prin cantitatea de octeți prelucrați pe secundă
- latența, care măsoară timpul scurs de la setarea circuitului până la obținerea secvențelor de ieșire
- puterea consumată măsurată în Wați.

În cazul implementărilor software, parametrii importanți sunt: consumul de memorie RAM, care reprezintă cantitatea de memorie necesară pentru o rulare a algoritmului, dimensiunea codului sursă, viteza de procesare și consumul de putere.

2.6. Managementul criptografic pentru protocoalele de securitate folosite în IoT

În acest subcapitol sunt prezentate aspecte specifice referitoare la managementul cheilor criptografice pentru următoarele protocoale utilizate în infrastrcuturi IoT:

- Protocolul BLE (Bluetooth Low Energy)
- Protocolul IEEE 802.15.4
- Protocolul Zigbee
- Protocolul LoRaWAN
- Protocolul Z-Wave
- Protocolul IEEE 802.11 - Wi-Fi Protected Acces
- Protocoalele TLS – Transport Layer Security

2.7. Managementul accesului și al identităților

Prin IoT s-a introdus conceptul prin care entitățile sunt interconectate. Pentru a realiza un context sigur în care acestea să comunice este nevoie, în primul rând, de un mecansim prin care să poată fi identificate. Noțiunea de identitate se referă la un set de informații utilizate pentru a identifica în mod unic o entitate într-un anume context. De exemplu, o persoană poate fi identificată la locul de muncă printr-un set de atribute precum numele, funcția ocupată, tipul funcției iar într-un magazin online de atribute precum nume și cont bancar.

Un sistem de management al accesului și al identităților (MAI) are în vedere ciclul de viață al identităților, care cuprinde operații de înregistrare, actualizare și revocare a acestora. În cadrul unui sistem, MAI trebuie să asigure trei servicii de securitate: autentificarea, autorizarea și auditul. De exemplu, în cazul unui operator care dorește acces la un serviciu, operația de autentificare constă în introducerea credențialelor pentru identitatea revendicată și operația de autorizare care verifică credențialele și ia decizia de a oferi sau nu accesul. Toate aceste operații sunt monitorizate și înregistrate de serviciul de audit.

2.7.1. Metode de autentificare

Metodele de autentificare se pot clasifica în funcție de credențialele folosite. Acestea pot fi de mai multe tipuri, astfel:

- username sau ID și o parolă.
- credențiale care se referă la ”ceva” care se deține. În cazul persoane acel ceva paote fi un generator de parole unice, un card, un token sau un smartphone, . În cazul dispozitivelor IoT, acel ”ceva” se referă la un secret stocat intern pe baza căruia, folosind un algoritm, se poate dovedi autenticitatea identității invocate.
- credențiale care se referă la ”ceea ce ești”. În cazul persoanelor reprezintă date biometrice iar în cazul dispozitivelor IoT sunt PUF-urile (Physical Unclonable Function). Acestea sunt obiecte fizice (dispozitive semiconductoare, microprocesoare) care oferă răspunsuri unice ce pot fi asimilate cu amprente digitale.
- credențiale legate de context. Acestea sunt folosite de obicei cu rol complementar. De exemplu, pentru persoane poate reprezenta locația GPS combinată cu informații despre timp iar în cazul dispozitivelor IoT, poate reprezenta caracteristici legate de locația geografică și de tehnologia de comunicație.

2.7.2. Metode de autorizare

În sistemele IoT se folosesc diferite tipuri de metode de autorizare, fiecare cu avantajele și dezavantajele lor. Acestea pot fi clasificate în funcție de modelul de control al accesului.

- DAC (Discreționary Acces Control) - proprietarul dispozitivului IoT hotărăște regulile de acces care pot restricționa perioada de timp în care se oferă accesul, operațiile disponibile și entitățile care au acces;
- MAC (Mandatory Acces Control) - autorizarea se oferă gradual în funcție de tipul de acces deținut;
- RBAC (Role Based Acces Control - există mai multe roluri cărora li se asigură permisiuni iar fiecărui utilizator îi este asignat unul sau mai multe roluri în funcție de responsabilități;
- ABAC (Attribute-Based Access Control - mai multă flexibilitate prin faptul că, în loc să definească un rol static, utilizează un set de politici pentru a acorda accesul.
- Cap-BAC (Capability-Based Access Control) este un model de control acces bazat pe tokene, care stochează dreptul de acces utilizatorilor care îl dețin;

Alte metode convenționale de autorizare sunt prezentate în: Lattice-Based Access Control, Context-Based Access Control, Chinese Wall Lattice Model, Identity-Based Access Control.

2.7.3. Clasificarea sistemelor de management al identităților

Sistemele de management al identităților (SMI) au evoluat odată cu tehnologia. În prezent se pot identifica cinci tipuri de astfel de sisteme [9]:

- izolate
- centralizate
- federative
- centrate pe utilizator
- autosuveran (self-sovereign).

2.8. Aplicabilitatea tehnologiei blockchain în IoT

2.8.1. Aspecte de securitate referitoare la tehnologia blockchain

Tehnologia blockchain are ca element central o bază de date de tip registru (ledger) alcătuit din blocuri înlănțuite (vezi Figura 2.2). Legătura dintre blocuri este realizată folosind funcții de tip hash, în sensul că un bloc conține hash-ul blocului anterior pe lângă informații legate de timp, tranzacții sau date. Datorită acestui model de proiectare se asigură proprietatea de imutabilitate, ceea ce înseamnă că, pentru a modifica informația dintr-un bloc, vor trebui modificate toate blocurile care îl succed. Acest lucru nu este ușor de realizat, deoarece registrul se stochează în mod distribuit de către membrii rețelei, care îl actualizează în mod continuu conform regulilor unui protocol de consens. Pentru a putea avea totuși succes este nevoie de consensul a peste 50% dintre membrii rețelei.

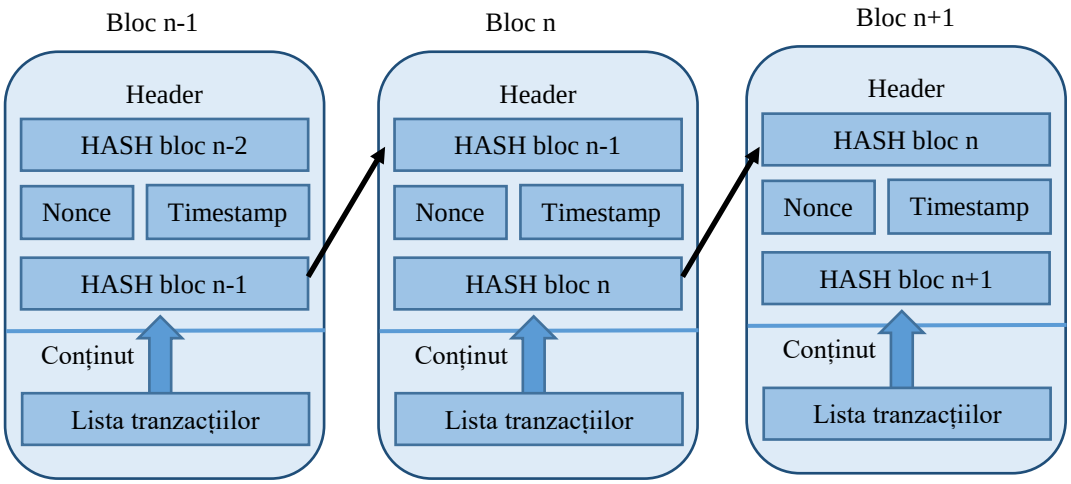

Figura 2.2 Arhitectura blockchain

Protocolul de consens este unul dintre elementele cheie care asigură securitatea în blockchain. Pe baza lui membrii rețelei validează introducerea unui bloc în lanțul deja existent. În prezent au fost propuse mai tipuri de astfel de protocoale, fiecare cu avantajele și dezavantajele lui. Acestea au următoarele caracteristici:

- Proof of Work (PoW) - cel mai cunoscut fiind folosit în Bitcoin și Ethereum. Ideea principală a acestui prototol este de a folosi puterea de calcul pentru a valida blocul

care se dorește a fi introdus. Pentru acest lucru, minerii care fac parte din utilizatorii rețelei încearcă să identifice un număr care, împreună cu datele efective din bloc, să aibă un hash cu anumite caracteristici. Acest lucru se poate realiza prin încercări succesive. Puterea de calcul este necesară pentru calculul funcției hash respective. Chiar dacă oferă un grad de securitate sporit, acest tip de protocol este mare consumator de energie.

- Proof of Stake (PoS). În acest caz minerii sunt aleși dintre membrii rețelei, care pun la dispoziție o parte din monedele virtuale pe care le dețin. Dacă aceștia nu validează corect, vor pierde monedele puse la dispoziție. Acest tip de protocol va fi implementat în curând și în Ethereum iar variante asemănătoare sunt folosite și în alte blockchain-uri, cum ar fi Elrond
- Proof of Capacity (PoC), unde minerii pun la dispoziția rețelei o anumită capacitate de stocare, ceea dovedește un anumit grad de interes pentru ca sistemul să funcționeze corect
- Proof of Autority (PoA), care se bazează pe reputația minerilor selectați pentru validare.

Un alt aspect important, care are influențe și asupra securității, este tipul de blockchain. Fiecare blockchain are reguli care pot permite participarea oricui în rețea sau pot limita participarea doar pe bază de permisiune. În funcție de aplicație aceste două tipuri pot oferi anumite avantaje. Primul tip permite accesul nelimitat în rețea, asigurând astfel o descentralizare și transparență totală. În acest caz nu există o autoritate centrală iar anonimitatea participanților este asigurată. Al doilea model este pretabil pentru a fi utilizat în cadrul organizațiilor unde anonimitatea membrilor nu este necesară. Acest model nu oferă decât o descentralizare parțială care poate oferi un anumit grad de securitate în cazul unui atac din exterior. Datorită numărului limitat de participanți această arhitectură oferă viteze mai mari și scalabilitate crescută. De altfel se poate asigura și o protecție superioară a datelor stocate în registru din moment ce accesul este restricționat.

2.8.2. Aplicații blockchain în IoT

Avantajele tehnologiei blockchain a deschis noi drumuri, ajutând la utilizarea acesteia în aplicații din multe domenii. Pe lângă domeniul financiar, consacrat prin aplicațiile de criptomonede, în [10] sunt menționate și alte arii de interes, precum:

- securitate cibernetică;
- aplicații din zona guvernamentală;
- sistem de înregistrare și management al proprietăților (case, terenuri) sau a valorilor (mașini, telefoane) ;
- managementul identităților;
- sistem de management al reputației;
- proprietate intelectuală;
- strângere de fonduri;

- sisteme eneregetice;
- aplicații IoT.

Aplicațiile IoT sunt clasificate în mai multe tipuri: asigurarea protecției cibernetice în sisteme energetice, asigurarea protecției cibernetice în sisteme de transport, asigurarea protecției cibernetice în sisteme aviatice, sisteme de siguranță alimentară, case inteligente, aplicații militare precum IoBT (Internet of Battle Things), sistem de management access sau de management chei publice.

3. Soluții blockchain pentru asigurarea securității IoT

În acest capitol sunt prezentate câteva soluții de utilizare a tehnologiei blockchain pentru asigurarea securității IoT. După ce se analizează avantajele și dezavantajele utilizării tehnologiei blockchain în IoT se propun două soluții de integrare IoT cu BC. Prima soluție prezintă un model de arhitectură fog computing, care integrează un blockchain, pentru a folosi proprietățile de securitate ale acestuia cu scopul de realiza o relație de încredere între membrii rețelei. Folosind blockchain-ul pentru stocarea cheilor de identitate se propune un protocol simplu și sigur de stabilire a cheilor de sesiune și autentificare. Soluția este evaluată din punctul de vedere al puterii de calcul și costurilor, fiind comparată cu o soluție clasică cu TLS și PKI. A doua soluție propune utilizarea tehnologiei FPGA pentru integrarea IoT cu BC. Astfel, se propun două arhitecturi FPGA de implementare a nodurilor de senzori cu rol dublu de noduri de senzori și noduri BC. Soluțiile propuse sunt implementate pe mai multe familii de circuite FPGA, cu consum și resurse diferite.

3.1. Analiza posibilității de utilizare a tehnologiei blockchain în IoT

Asigurarea securității în infrastructurile IoT necesită tehnologii adecvate. Tehnologia blockchain (BC) prezintă anumite avantaje care o pot califica în acest sens. În continuare sunt prezentate anumite caracteristici ale acesteia care dovedesc posibilitatea integrării în infrastructura IoT:

- *descentralizarea* oferă posibilitatea ca tranzacțiile să nu fie validate de o entitate centrală, care ar putea fi suprasolicitată în cazul unui număr mare de tranzacții sau în cazul unor atacuri de tip Denial of Service (DoS);
- *imutabilitatea* care se manisfestă prin faptul că tranzacțiile stocate în registrul BC nu pot fi modificate. Astfel, dispozitivele IoT au posibilitatea de a verifica ușor și sigur datele stocate;
- *reziliența datelor* este asigurată prin faptul că nodurile sunt în posesia unei copii a bazei de date blockchain;
- *suport criptografic*, deoarece tehnologia blockchain se bazează pe funcții care pot asigura servicii de confidențialitate, integritate și autentificare;
- *încredere* într-un mediu eterogen, cum este cel al IoT, prin faptul că poate oferi încredere între membrii rețelei fără a fi nevoie de o autoritate centrală;

- *audit* - tehnologia blockchain oferă posibilitatea de înregistrare a operațiilor în registru în mod sigur și imutabil, acestea putând fi vizualizate de toți membrii rețelei.

În același timp, caracteristicile unei infrastructuri IoT sunt compatibile cu tehnologia blockchain. Pentru a asigura o descentralizare cât mai bună este necesar de un număr mare de noduri, aspect pe care infrastructurile IoT îl îndeplinesc. Deoarece într-o infrastructură IoT pot fi active la un moment dat de timp un număr mare de noduri reprezintă un avantaj pentru o infrastructură blockchain care are nevoie de entități care să valideze tranzacțiile.

Pe de altă parte, integrarea celor două tehnologii vine și cu anumite provocări. Iplementarea tehnologiei blockchain necesită anumite *resurse* de care nu toate nodurile IoT dispun (capacitate de calcul redusă, capacitate de stocare, număr de tranzacții limitate).

În prezent nu sunt rezolvate toate problemele care ar putea apărea în cazul integrării celor două tehnologii. Provocarea constă în identificarea unor arhitecturi care să integreze cele două tehnologii pentru a profita de avantajele pe care tehnologia blockchain le aduce securității, dar care să diminueze pe cât posibil neajunsurile implementării acestuea în IoT.

3.2. Soluție de integrare IoT- BC

3.2.1. Caracteristici arhitecturale ale tehnologiei fog computing

Aplicațiile IoT folosesc date achiziționate de la senzori. De cele mai multe ori volumul de date colectat de senzori este foarte mare. Multe dintre aceste date nu sunt folosite în mod direct, fiind necesară prelucrarea acestora. Dat fiind faptul că dispozitivele prin intermediul cărora se colectează datele nu dispun de suficientă putere de calcul s-a optat pentru colectarea acestora în Cloud, unde se prelucrează și se distribuie către aplicații specifice. De cele mai multe ori nodurile de senzori sunt distribuite pe o arie geografică întinsă sau se află în locații unde nu sunt disponibile rețele de comunicații care să suporte transferul unui volum mare de date. În aceste cazuri pot apărea întârzieri mari în transferul datelor sau pierderi semnificative de informații, ceea ce poate influența în mod negativ calitatea serviciilor. Una dintre soluțiile care pot rezolva aceste probleme este crearea unui nivel intermediar de dispozitive între nodurile de senzori și Cloud. Această arhitectură este numită *fog computing*. Fog computing este o arhitectură descentralizată, specifică arhitecturilor IoT, care portează o parte din serviciile disponibile în Cloud către marginile (în engleză edge) rețelei. Acest tip de arhitectură are avantajul de a porta resursele de calcul specifice din Cloud aproape de locul unde datele sunt achiziționate. În acest fel, se pot folosi algoritmi complexi de prelucrare a datelor sau inteligența artificială pentru sortarea eficientă a datelor pe dispozitive situate aproape de locul unde acestea sunt create. Astfel eficiența este îmbunătățită prin reducerea volumului de date transferat către Cloud.

În [11], NIST a descris această arhitectură ca fiind un ecosistem bazat pe Cloud în care fog computing deservește dispozitivele IoT finale. Arhitectura este construită pe patru niveluri. Primul nivel este alcătuit din dispozitivele IoT finale, reprezentate de senzori și actuatori. Datele provenite de la aceștia sunt colectate prin intermediul dispozitivelor de nivelul următor, denumit *aburi* (în engleză *mist*). Pe acest nivel se regăsesc diferite noduri specializate de

senzori care sunt implementate folosind în special microcontrolere sau microcomputere. Acestea dispun de putere de calcul redusă. Acest nivel reprezintă legătura cu nivelul următor – fog computing, care este alcătuit din noduri cu putere de calcul, capacitate stocare și resurse de rețea mult superioare dispozitivelor de pe nivelul mist. Acestea pot fi dispozitive de rețea puternice precum gateway, rutere, sau calculatoare care dispun de putere mare de calcul, precum serverele și mini centre de prelucrare a datelor – *cloudlets*.

3.2.2. Securizarea unei arhitecturi fog computing folosind BC

Tehnologia blockchain are la dispoziție resurse să asigure servicii de securitate. Deoarece este alcătuită pe o infrastructură descentralizată, are premise să asigure securitate și într-o arhitectură de tip fog computing. Securitatea în BC este asigurată de noduri care decid, în baza unui protocol de consens, datele care se înregistrează în registrul BC. Cu cât este mai mare numărul de noduri și cu cât sunt mai distribuite geografic pe o arie mai mare cu atât securitatea asigurată de rețea este mai puternică. Astfel de caracteristici pot fi asigurate de o arhitectură de tip fog computing.

În arhitecturile curente securitatea este asigurată centralizat prin intermediul infrastructurilor cu chei publice. În acest fel, o entitate de încredere emite - semnează certificate digitale pentru toți membrii rețelei. Aceste certificate digitale conțin o pereche de chei asimetrice care asigură protecția criptografică în diferite servicii de securitate. Există totuși incertitudinea că aceste sisteme sunt potrivite pentru infrastructuri IoT. Un prim argument ar fi că numărul foarte mare de dispozitive IoT nu ar putea fi deservit prompt și eficient de o astfel de infrastructură. Un alt argument este cel legat de capacitatea de procesare a unora dintre dispozitivele IoT, care nu suportă o astfel de tehnologie. Această incertitudine este alimentată și de faptul că infrastructura PKI este centralizată și deci insuficient de transparentă. Blockchain are capacitatea de a rezolva aceste probleme.

3.2.3. Descriere arhitectură propusă

Soluția propusă de integrare a tehnologiei blockchain într-o arhitectură de tip fog computing își propune să asigure un context de securitate propice asigurării serviciilor de autentificare mutuală între nodurile IoT si a serviciilor de confidențialitate și integritate a comunicațiilor. Provocarea, în acest caz, este de a identifica modalitățile prin care cele două tehnologii se pot integra astfel încât să aducă beneficii superioare soluțiilor clasice care folosesc PKI. Arhitectura propusă și rezultatele experimentale obținute au fost publicate în lucrarea [12].

Utilizarea BC într-o infrastructură PKI poate întâmpina anumite probleme. Prima dintre ele se referă la resursele necesare pentru asigurarea funcțiilor de bază din BC. Protocoale de consens precum PoW necesită putere de calcul semnificativă. Soluțiile de înlocuire a acestui protocol cu altele mai puțin consumatoare de energie nu rezolvă în totalitate problema. Este necesară energie considerabilă și pentru a asigura capacitățile de comunicare între noduri. În soluția propusă în Figura 3.1, funcțiile nodurilor BC sunt implementate pe noduri specifice

arhitecturii fog computing, care dețin suficientă putere de calcul. Astfel, dispozitivelor din cloud sau de pe nivelul FOG li se asignează nodurile BC care îndeplinesc funcțiile de bază, cum ar fi minarea și stocarea registrului. Dispozitivelor de pe nivelul FOG cu resurse limitate sau dispozitivelor de pe nivelul MIST li se asignează funcții precum validarea sau interogarea registrului BC. Celalte noduri pot participa la rețeaua BC din rol de utilizatori, care au acces direct la blockchain sau prin intermediul unui nod cu care are stabilită deja o conexiune de încredere.

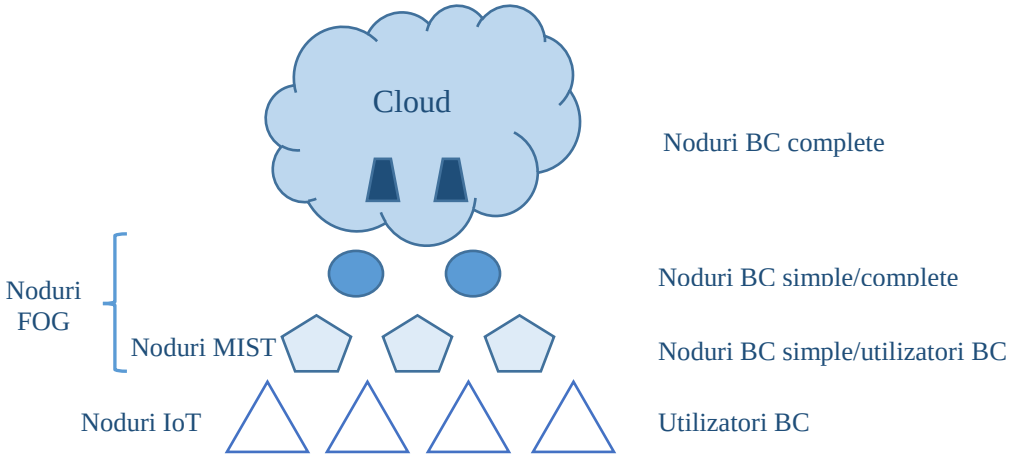

Figura 3.1 Arhitectura fog computing - blockchain

Pentru a valida soluția propusă am realizat o implementare a unui contract inteligent prin care un nod IoT poate interacționa cu BC. Implementarea și testarea au fost realizate folosind mediul Remix IDE and Ganache, care simulează blockchain-ul Ethereum. Funcțiile implementate în contractul inteligent au fost următoarele:

- Înregistrare nod IoT – înregistrează informații referitoare la un nod IoT : ID device, cheie de identitate și adresă IP;
- Modificare nod IoT – modificare adresă IP sau cheie de identitate;
- Citire informații nod IoT – citire adresă IP sau cheie de identitate;

În Tabelul 3.1 sunt prezentate costurile implementării contractului inteligent în blockchain-ul Ethereum. Din rezultatele prezentate în tabel se poate observa că cele mai mari costuri sunt necesare pentru crearea contractului în BC. Costul de înregistrare a unui nod este de 10 cenți iar pentru modificare costul este redus la jumătate. Pentru citirea datelor din BC, operații care se desfășoară în mod frecvent, costurile sunt zero. Estimările de preț prezentate în tabel au la bază prețul mediu din ultimul an de zile (16.05.2022 -16.05.2023).

Tabelul 3.1 Costuri operații contract inteligent

Tipul tranzacției	**Ethereum gas**	**Pret gas (gwei)**	**ETH**	**Preț ETH ($)**	**Preț ($)**
Creare contract	267.177	31,42	0,000267177	1.535,31	0,41
Înregistrare dispozitiv IoT	65.672	31,42	0,000065672	1.535,31	0,10
Modificare dispozitiv IoT	29.472	31,42	0,000029472	1.535,31	0,05

Costurile necesare implementării și întreținerii unei astfel de soluții au fost calculate pentru un număr de 1000 de dispozitive IoT și comparate cu cele ale soluțiilor clasice cu certificate PKI. Pentru comparație, am folosit informațiile disponibile pe site-ul clickSSL.net [13]. În Tabelul 3.2 sunt prezentate costurile totale pentru cele două soluții. În cazul soluției cu BC sunt incluse costurile cu înregistrarea dispozitivelor IoT și costurile anuale cu înlocuirea cheii de criptare. În cazul soluției PKI sunt incluse costurile necesare emiterii de certificate digitale pentru dispozitivele IoT pe perioade diferite de timp.

Tabelul 3.2 Comparație costuri

Perioadă	Preț($) 1 an	Preț($) 3 ani	Preț($) 5 ani
Soluție propusă	101,23	191,73	1236,03
Soluție PKI	14.000	36.000	50.000

3.3. Protocol de negociere chei criptografice pentru o arhitectură IoT-BC

3.3.1. Soluția propusă

Soluția propusă folosește tehnologia distribuită blockchain ca o ancoră de securitate pentru a stabili relații de încredere între membrii unei rețele IoT și apoi, prin intermediul unor schimburi de mesaje, stabilește chei de criptare și autentificare. În Figura 3.2 este prezentată arhitectura soluției. Un nod IoT care dorește să inițieze o sesiune de comunicație cu alt nod are nevoie de cheia acestuia de identitate. Această cheie de identitate o poate găsi într-un blockchain în care nodurile au fost înregistrate în prealabil. Arhitectura prezintă două tipuri de noduri. Nodurile IoT de tip *F*, care au la dispoziție suficiente resurse ca să poată participa în blockchain, vor interoga registrul BC pentru a afla cheia de identitate a corespondentului. Acestea pot fi noduri de tip FOG sau MIST cu roluri de noduri cu funcții active în BC sau doar utilizatori ai BC. Al doilea tip de noduri sunt cele de tip *N*, care nu dispun de suficiente resurse pentru a accesa BC dar folosesc un nod de tip F pentru a accesa registrul BC.

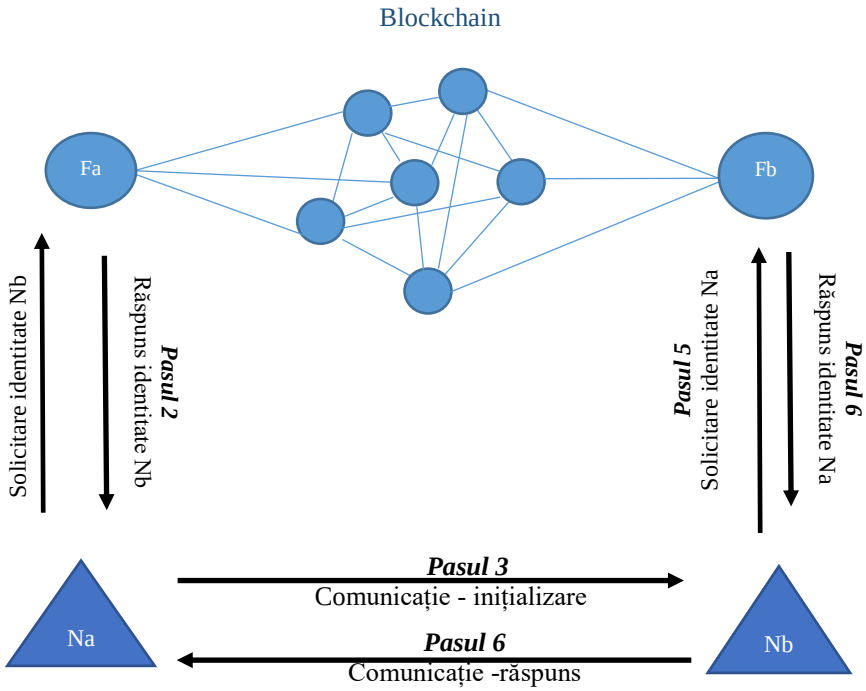

Figura 3.2 Arhitectură protocol de negociere

Pentru a se asigura prin intermediul protocolului generarea cheilor sesiune în contextul de securitate dorit este necesară realizarea unor operații în prealabil. În primul rând nodurile IoT trebuie să fie înregistrate în blockchain. Acest lucru se poate realiza folosind un contract inteligent prezentat anterior. Acesta trebuie să conțină cheia de identitate a nodului și câteva informații necesare identificării acestuia: Id, adresă IP etc. Pașii necesari pentru înregistrarea unui nod în BC sunt următorii:

- nodul IoT generează o pereche de chei de identitate folosind curba eliptică x25519;
- nodul IoT apelează un contract inteligent prin intermediul căruia înregistrează în BC partea publică a cheii de identitate împreună cu informațiile necesare identificării. Această operație poate fi realizată și de către proprietarul nodului IoT, din postură de utilizator BC;
- Fiecare nod va fi identificat printr-un Id unic, care reprezintă hash-ul cheii publice de identitate, calculată cu funcția SHA256;
- Fiecărui nod de tip N i se va transmite printr-o metodă sigură cheia de identitate a nodului F la care este alocat.

În funcție de tipul nodurilor protocolul are un număr diferit de pași. În cazul nodurilor de tip N, se mai adaugă doi pași, în care acesta solicită și primește de la nodul de tip F, la care este acesta este afiliat, cheia de identitate a corespondentului.

Funcțiile criptografice utilizate în cadrul protocolului sunt următoarele:

- generarea cheilor de autentificare și criptare – funcția HKDF (cheie, salt). Cheia reprezintă secretul generat folosind curba eliptică x25519 din partea privată a cheii de identitate a expeditorului și partea publică a cheii destinatarului. Saltul

reprezintă un număr aleatoriu. Pentru a asigura generarea unei chei diferite, de fiecare data saltul va fi diferit;
- criptarea mesajelor se realizează cu algoritmul criptografic AES în modul CBC cu cheie de lungime 32 de octeți și vector de inițializare de 16 octeți;
- autentificarea mesajelor se realizează cu algoritmul HMAC SHA 256. Fiecare mesaj este autentificat. Destinatarul va verifica autenticitatea mesajului la primire.

3.3.2. Analiza soluție

3.3.2.1. Analiză de securitate

Toate informațiile sensibile care ar putea oferi posibilități de inițiere a unor atacuri sunt criptate și toate mesajele sunt autentificate. Cheile de criptare și autentificare folosite sunt diferite pentru fiecare mesaj în parte datorită utilizării la generare a unui salt diferit și aleator. Operațiile criptografice sunt realizate folosind algoritmi puternici și siguri: AES 256 pentru criptarea simetrică, HMAC SHA 256 pentru integritate și autentificare, HKDF pentru generarea cheilor și secretelor și curba eliptică x25519 pentru criptarea asimetrică.

3.3.2.2. Analiză performanțe

Pentru a pune în evidență consumul mic de energie necesar negocierii unei chei de sesiune folosind protocolul prezentat, am realizat o comparație cu protocoalele TLS 1.2 și TLS 1.3. Implementarea a fost realizată pe platforma B-L475E-IOT01A, dezvoltată de STMicroelectronics.

Tabelul 3.3 Consum energie funcții criptografice

Funcție criptografică	Energie consumată (Wh)
ECDH semnare (32 octeți)	206,955
ECDH verificare semnătură (32 octeți)	131,5872
x25519 generare secret	15,33168
x25519 generare cheie	15,22872
HMAC SHA256 (32 octeți)	0,0575316
SHA 256 (334 octeti)	0,074646
HKDF SHA256	0,177021
ENC AES CBC 128 (16 octeți)	0,01340352
DEC AES CBC 128 (16 octeți)	0,0132561
ENC AES GCM 128 (16 octeți)	0,0946737
DEC AES GCM 128 (16 octeți)	0,0815346
ENC AES CBC 256 (32 octeți)	0,0231984
DEC AES CBC 256 (32 octeți)	0,0229086

Pentru ca rezultatele să nu fie influențate de mediul de transmisie am luat în considerare doar funcțiile criptografice implicate în proces. Implementarea acestora a fost realizată folosind librăria software wolfSSL. Energia consumată a fost calculată prin înmulțirea curentului, măsurat pe timpul execuției fiecărei funcții, cu timpul necesar execuției acelei funcții și cu

tensiunea de alimentare (5V). Curentul microcontrolerului a fost măsurat pe pinul JP5 al plăcii electronice.

În Tabelul 3.3 sunt prezentate rezultatele obținute în urma implementării funcțiilor criptografice folosite de protocoloalele comparate. După cum se poate observa din tabel, energia necesară execuției funcțiilor de semnare și verificare a semnăturii este mult mai mare decât cea necesară operațiilor de criptare sau de calcul al hash-ului.

Energia necesară întregului proces de generare a cheilor, corespunzătoare fiecărui nod paticipant, este prezentată în Tabelul 3.4.

Tabelul 3.4 Energia consumată în timpul negocierii cheii de sesiune

Protocol	Nod	Energie consumată (Wh)
TLS 1.2	Client/Server	647,82
TLS 1.3	Client/Server	740,88
Soluția prezentată	Na	46,80
	Nb	46,98
	Fa	15,84
	Fb	15,84

Așa cum se poate observa, soluția propusă consumă mult mai puțină energie decât protocoalele TLS. Acest lucru se datorează faptului că este mult mai simplă decât soluția TLS, care a fost proiectată pentru a satisface situații mult mai diverse și complexe. În cazul unor aplicații IoT de complexitate redusă, soluția propusă poate asigura un grad de securitate ridicat, folosind resurse sensibil mai puține.

3.4. Integrarea nodurilor IoT în BC folosind FPGA

3.4.1. Soluție de implementare a nodurilor de senzori folosind FPGA pentru integrarea cu BC

În studiul publicat în [14] am descris o soluție de implementare a unui nod de senzori folosind circuite FPGA pentru a fi integrat în BC. Un nod implementat cu circuite FGPA poate asigura interfețe de comunicare cu senzorii de la care colectează date dar și interfețe specifice unui nod din blockchain, îndeplinind astfel ambele roluri.

În funcție de resursele hardware ale circuitului FPGA, nodul poate îndeplini diverse roluri în BC. În continuare vor fi prezentate două arhitecturi specifice circuitelor FPGA și funcțiile specifice pe care acestea le pot îndeplini

- ***Arhitectură clasică FPGA***

În această arhitectură propusă în Figura 3.3 toate blocurile componente sunt implementate folosind porțile logice puse la dispoziție de FPGA. Această arhitectură are avantajul că este foarte flexibilă și se poate adapta, pe de o parte la tipul și numărul de senzori pe care trebuie să îi deservească, dar poate avea implementate și blocuri de prelucrare a datelor specifice blockchain. În acest caz, se poate obține cel mai bun consum de energie pentru implementarea acestor funcționalități.

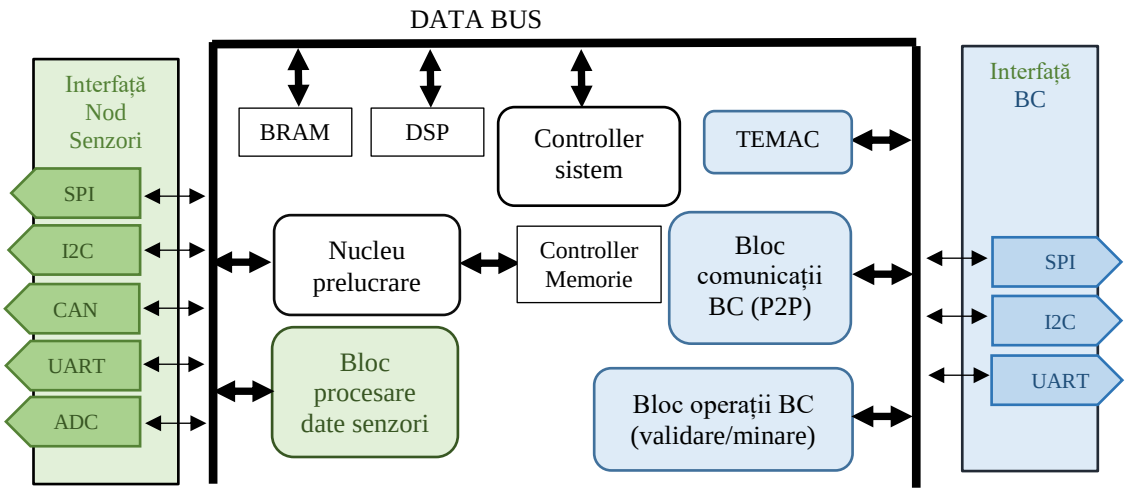

Figura 3.3 Arhitectură clasică FPGA pentru un nod de senzori

- ***Arhitectură de tip SoC FPGA***

Acest tip de arhitectură FPGA dispune de module de procesare dedicate incluse în blocul de procesare hardware. Doar familiile de circuite evoluate, precum Zynq-700, Zynq Ultra Scale+, Cyclone V dispun de arhitecturi de acest tip. Aceste circuite fac parte din categoria System on a Chip (SoC) care integrează unități de procesare împreună cu câteva tipuri de controlere, putând fi utilizate pentru implementarea blocurilor de monitorizare, de memorie și a modulelor dedicate de criptare. Pentru a funcționa corespunzător, acest bloc are nevoie de o memorie RAM externă, care este mare consumatoare de energie. Aceste avantaje ale arhitecturilor de tip SoC măresc considerabil capacitatea și viteza de procesare, recomandându-le pentru aplicații care necesită viteze mari de transfer, capacitate mare de procesare sau achiziție date de la multe noduri de senzori. Ținând cont de aceste aspecte, se poate afirma că această arhitectură se poate folosi pentru noduri hibride care să integreze atât funcționalități de nod BC cât și de nod senzori IoT.

În arhitectura propusă în Figura 3.3 interfața cu senzorii este implementată ca în varianta anterioară, folosind porți logice. Interfața BC se implementează folosind blocurile dedicate din blocul de procesare hardware. În acest fel se asigură, pentru această interfață, resursele de care are nevoie.

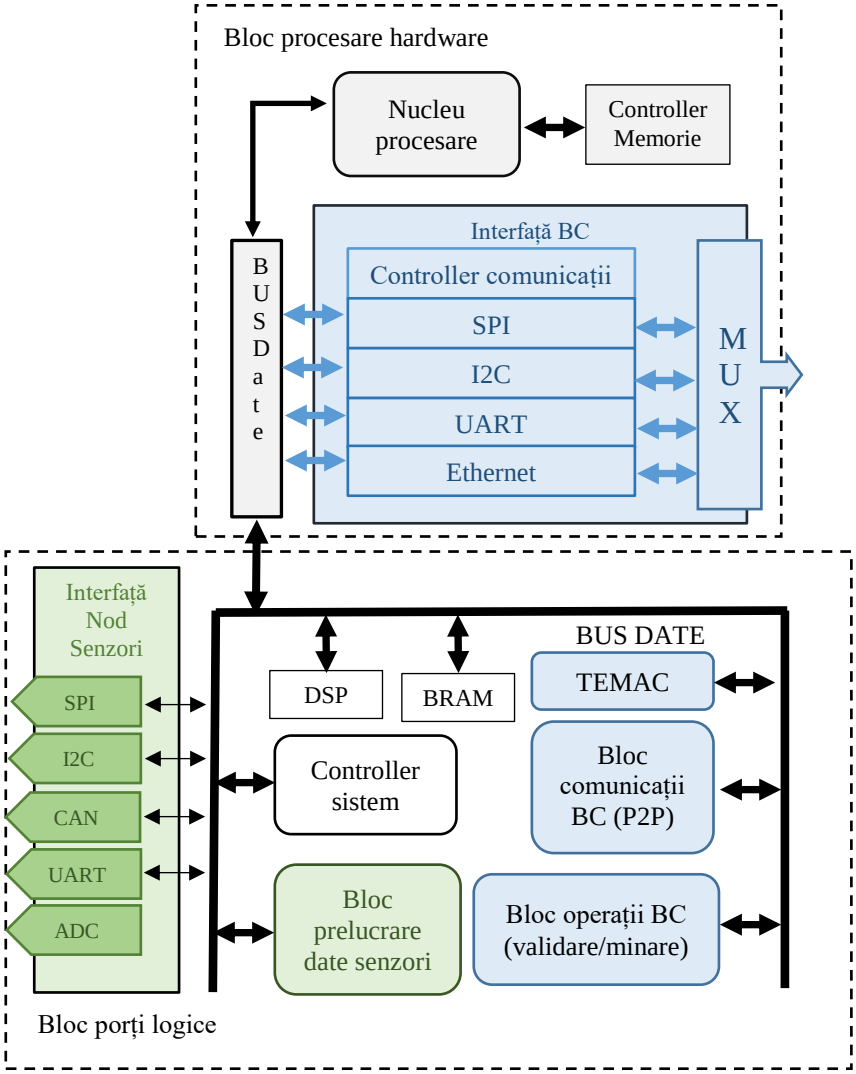

Figura 3.4 Arhitectura de tip SoC FPGA pentru un nod de senzori

3.4.2. Descriere experimente și prezentare rezultate

Soluțiile de implementare a unor noduri IoT folosind circuite FPGA au fost analizate în studiul pe care l-am publicat în lucrarea [14]. Experimentele s-au desfășurat în două etape. În prima etapă am implementat trei dintre cele mai utilizate blocuri de comunicații: SPI, I2C și TEMAC. Scopul experimentului a fost de a identifica necesarul de resurse utilizate pentru implementarea acestor blocuri necesare oricărui nod IoT. În acest mod se pot calcula resursele disponibile pentru implementarea celorlalte funcții specifice BC. În a doua etapă a experimentului am realizat o implementare optimizată a funcției SHA256, folosită intensiv pentru operațiile de minare în arhitecturile blockchain care utilizează protocolul de consens PoW.

În Tabelul 3.5 sunt prezentate resursele necesare implementării celor mai utilizate module de comunicații exprimate în Look –Up-Tables (LUTs). Aceste informații sunt foarte utile în procesul de proiectare a unui nod IoT și selectare a tipului de circuit FPGA care să acopere nevoile de comunicații ale nodului.

Tabelul 3.5 Resurse blocuri de comunicație

Bloc implementat	Resurse consumate (LUTs)
TEMAC	1256
SPI	164
I2C	216

În partea a doua a experimentelor am realizat o implementare optimizată din punct de vedere al timpului de execuție al funcției SHA 256. Modulul SHA 256 a fost implementat în 64 de ceasuri (clocks) pentru un bloc de intrare de 512 de biți, folosind doar 1152 LUTs.

Această implementare a funcției SHA 256 a fost realizată folosing circute din mai multe familii FPGA. Implementările au fost realizate atât pe circuite cu resurse limitate cât și pe circuite cu mai multe resurse disponibile, din cadrul aceleiași familii.

Din rezultatele prezentate în Tabelul 3.6 se poate deduce că și circuitele cu resurse foarte puține și cu consum mic de putere pot fi utilizate pentru implementarea funcției SHA 256 la o viteză acceptabilă. De asemenea, dacă nodul este folosit intens pentru validare și nodul este conectat la rețea, se poate folosi un circuit de tip Virtex 7 pentru care s-au obținut viteze apropiate de 1 Tbit/s. În cazul în care consumul de putere este important, se poate alege o variantă optimă din punct de vedere al raportului viteză – consum. În acest caz cele mai bune rezultate au fost obținute de circuitul FPGA XC7A12T din familia Artix 7 și de circuitul XC7Z007S din familia Zynq 7000 pentru soluții de tip FPGA SoC.

Tabelul 3.6 Rezultatele implementării funcției SHA 256 pe diferite circuite FPGA

Familie - circuit FPGA	Tip circuit	LUTs disponibile	Instanțe SHA256	Frecvență maximă (MHz)	Viteză (Gbit/s)	Curent Iccq (mA)
Spartan 6 – XC6SLX9	FPGA	5720	3	69,46	1,66	4,9
Spartan 6 – XC6SLX150T	FPGA	92152	78	69,46	43,44	63
Artix 7 – XC7A12T	FPGA	8000	5	138,7	5,5	51
Artix 7 – XC7A200T	FPGA	134600	115	138,7	127,6	268
Kintex 7 – XC7K70T	FPGA	41000	34	151,5	41,2	208
Kintex 7 – XC7K480T	FPGA	298600	257	151,5	311,4	840
Virtex 7 – XC7V585T	FPGA	364200	314	196,1	492,6	1597
Virtex 7 – XC7VX1140T	FPGA	712000	473	196,1	966,3	3698
Zynq 7000 – XC7Z007S	FPGA SoC	14400	12	138,7	13,3	172
Zynq 7000 – XC7Z020	FPGA SoC	53200	46	138,7	51,3	437
Zynq 7000 – XC7Z030	FPGA SoC	76600	66	151,5	79,9	437
Zynq 7000 – XC7Z100	FPGA SoC	277400	240	151,5	290,8	1095

4. Sursă de entropie pentru utilizare în aplicații IoT

În acest capitol propun un model de sursă de entropie specifică mediului IoT, care își extrage datele de la senzori de mișcare. Soluția propusă este supusă unei metodologii complexe de testare și evaluare, cu scopul de a identifica nivelul de entropie generat în diferite scenarii de utilizare, de a analiza stabilitatea sursei pe termen lung și în cazul unor atacuri pasive și active. De asemenea, s-a încercat identificarea parametrilor pentru optimizarea performanțelor sursei de entropie în termeni de putere consumată și viteză de generare.

4.1. Sursă de entropie cu date extrase de la senzori

O soluție care poate să îndeplinească cerințe specific unei platforme IoT este o sursă de entropie care își extrage sursa de zgomot din datele colectate de la senzori. Această soluție poate fi implementată pe orice nod de senzori independent de platforma de achiziție a datelor de la senzori. Singurele aspecte de care trebuie ținut seama sunt cele referitoare la tipurile și caracteristicile senzorilor. Dat fiind faptul că folosește date de la senzori, se poate spune că folosește în mare parte resursele deja existente pe platformă. Nu în ultimul rând, această soluție oferă acces total la sursa de zgomot care poate fi analizată în detaliu.

În Figura 4.1 este prezentată sursa de entropie cu senzori pe care am propus-o în lucrarea [7]. În această arhitectură sursa de zgomot poate prelua datele de la următorii senzori de mișcare: accelerometru, magnetometru și giroscop. În urma analizei entropiei emise au fost aleși cei mai puțin semnificativi 8 biți din secvențele generate de senzori. Aceste date sunt convertite în format digital după care sunt prelucrate folosind funcția SHA256. Aceasta are rolul de a concentra entropia și de a uniformiza datele de ieșire. Sursei de entropie i se aplică în mod continuu teste de sănătate pentru detecția unor eventuale erori.

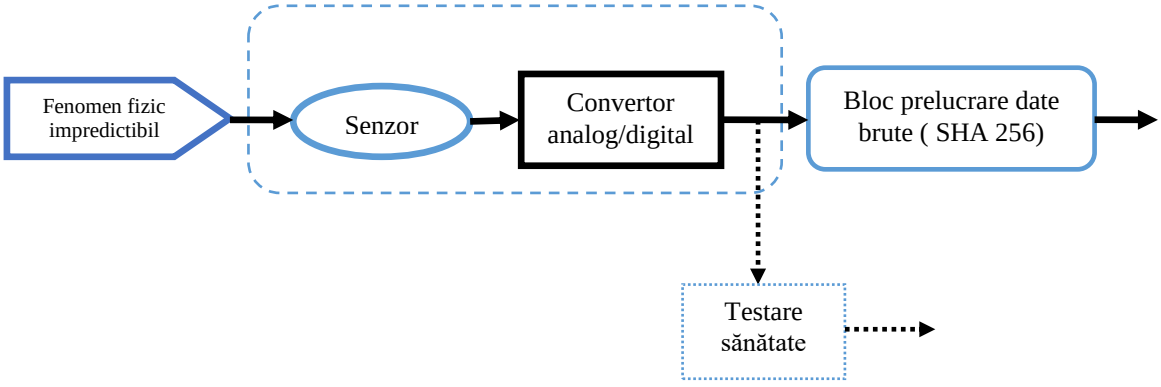

Figura 4.1 Arhitectură sursă de entropie cu senzori

4.2. Analiză, testare și validare sursă de entropie

4.2.1. Metodologie de estimare a entropiei

Metodologia de estimare a entropiei soluției propuse a fost analizată pe baza recomandărilor NIST din [15]. Aceasta presupune o estimare inițială a entropiei sursei de zgomot iar ulterior o estimare a ieșirii sursei de entropie, folosind date colectate imediat după

restartarea acesteia. Dacă estimările obținute la restart nu sunt mai mici decât jumătatea valorii entropiei estimate inițial, entropia finală a sursei va fi minimul valorilor estimate anterior, și anume: H_{init} – entropia inițială, H_{linie} și $H_{coloană}$ entropiile estimate din datele colectate la restart. Entropia se estimează folosind formula (4.2) a entropiei minime.

Pentru a estima entropia inițială H_{init}, se folosește un număr de 1.000.000 de secvențe colectate direct de la sursa de zgomot. Lungimea secvențelor se poate afla în intervalul de la 1 la 8 biți. Pentru secvențe mai mari de 8 biți se recomandă utilizarea unor metode de truncare a secvenței de ieșire astfel încât să fie utilizați biții care pot furniza cea mai mare entropie. În cazul unei surse bazată pe senzori de mișcare cea mai mare impredictibilitate o oferă biții cei mai puțini semnificativi. În lucrarea [8] am demonstrat acest aspect.

În funcție de tipul datelor pe care sursa le poate genera, entropia se evaluează diferit. În cazul în care sursa poate genera date independente și distribuite uniform, se aplică două tipuri de teste statistice: teste de permutare și teste chi-square. Dacă cele două tipuri de teste sunt trecute cu succes, se consideră că sursa generează entropie maximă. Pentru cazul în care datele nu sunt independente sau distribuite uniform se aplică metoda cu estimatori. Acestă metodă este utilizată și în cazul sursei de entropie bazată pe senzori. Astfel, entropia datelor colectate se calculează folosind un număr de 10 estimatori diferiți, care analizează diferite proprietăți statistice. În final se ia în considerare cea mai mică valoare a entropiei obținute de estimatori.

În cazul în care entropia finală nu este maximă, se poate folosi o funcție de condiționare (vezi Figura 4.2). În acest scop am folosit funcția SHA 256 care calculează un rezumat dintr-un număr de secvențe care conțin 512 biți de entropie. În acest caz funcția SHA 256 are rolul de concentrator de entropie. Numărul de secvențe folosite la intrarea funcției se calculează în funcție de numărul N_e de biți de entropie conținuți într-o secvență.

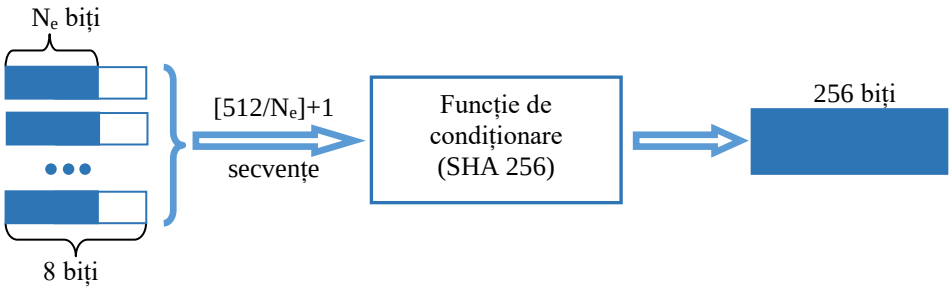

Figura 4.2 Metoda de uniformizare a entropiei

4.2.2. Metodologie de analiză, testare și validare a sursei de entropie

Sursa de entropie trebuie să fie capabilă să asigure un anumit nivel de entropie în orice condiții de utilizare. Din acest motiv, analiza unei surse de entropie trebuie să ia în considerare aspecte multiple, care țin de fenomenul fizic ce stă la baza sursei de zgomot și de stabilitatea sursei în timp și în diferite condiții de utilizare. În continuare, voi prezenta o metodologie originală și completă de testare și validare a surselor de entropie bazate pe senzori, care are drept reper recomandările NIST din [15]. În Figura 4.3 este prezentat un sumar al metodologiei de analiză și evaluare a sursei de entropie bazată pe senzori de mișcare.

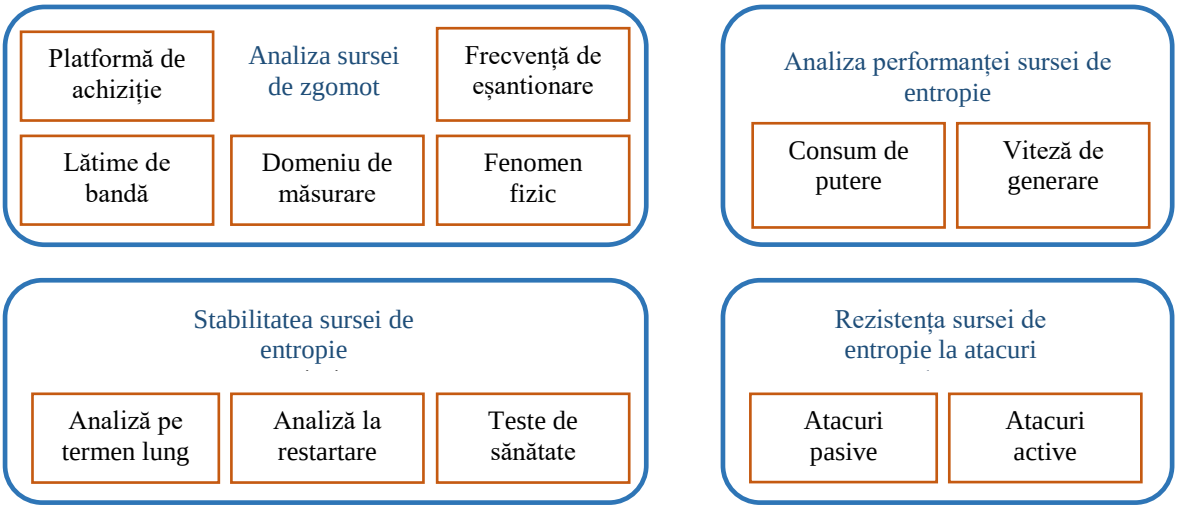

Figura 4.3 Metodologie de analiză și evaluare sursă de entropie

4.2.2.1. Analiza sursei de zgomot

În analiza sursei de zgomot am plecat de la modelul constructiv al acesteia, prezentat în Figura 4.4 Model sursă de zgomot, încercând să se identific influența fiecărui element în valoarea entropiei datelor de ieșire.

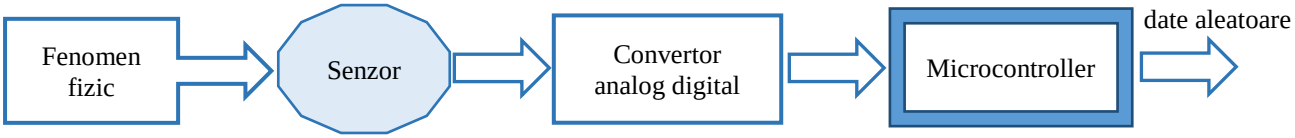

Figura 4.4 Model sursă de zgomot

Experimentele au fost efectuate folosind trei tipuri diferite de platforme, pentru a putea capta cinci tipuri de mișcări.

- Platforma 1 este o placă electronică de tip B-L475E-IOT01A, folosită ca nod de senzori destinat aplicațiilor IoT;
- Platforma 2 este alcătuită dintr-o placă de achiziție de tip Arduino UNO și un senzor de miscare de tip MPU 9250, care conține un accelerometru, un gisroscop și un magnetometru
- Platforma 3 folosește senzorii de mișcare cu care este dotat un telefon mobil. În cadrul experimentelor am captat date de la senzori cât timp senzorii au fost supuși la cinci tipuri de mișcări.

Rezultatele experimentelor se regăsesc în Tabelul 4.1 pentru accelerometre, în Tabelul 4.2 pentru giroscoape și în Tabelul 4.3 pentru magnetometre. Câteva observații se pot extrage din analiza acestor rezultate:

- indiferent de tipul de senzor, datele de ieșire ale acestuia conțin entropie. Cu cât mișcarea aplicată senzorilor de tip accelerometru și giroscop este mai amplă sau mai rapidă, cu atât valoarea entropiei este mai mare;

- entropia este obținută chiar și când nu se aplică mișcare suplimentară asupra senzorilor;
- o mișcare repetitivă nu influențează în mod negativ entropia, chiar din contră obține cele mai bune rezultate.

Tabelul 4.1 Influența tipului de mișcare asupra entropiei accelerometrelor

Tip mișcare / Platformă	Static (fără mișcare)	Autoturism în mișcare	Rotație	Alergare	Mers
Platforma 1	0,1997	0,7424	-	-	-
Platforma 2	0,2210	0,6846	0,8544		
Platforma 3	-	0,5019	-	0,4021	0,2905

Tabelul 4.2 Influența tipului de mișcare asupra entropiei giroscoapelor

Tip mișcare / Platformă	Static (fără mișcare)	Autoturism în mișcare	Rotație	Alergare	Mers
Platforma 1	0,0551	0,4147	-	-	-
Platforma 2	0,1207	0,4348	0,8299	-	-
Platforma 3	-	0,3598	-	0,2942	0,2335

Tabelul 4.3 Influența tipului de mișcare asupra entropiei magnetometrelor

Tip mișcare / Platformă	Static (fără mișcare)	Autoturism în mișcare	Rotație
Platforma 1	0,4368	0,5690	-
Platforma 2	0,0025	0,0049	0,1566

- *Domeniul de măsurare și lățimea de bandă*

Al doilea factor care poate influența valoarea entropiei este configurarea parametrilor senzorului utilizat. În acest scop au fost studiate în detaliu posibilitățile de parametrizare a senzorilor de mișcare și s-a constatat că anumiți parametri influențează nivelul de entropie iar alții nu au efect asupra acesteia.

Rezultatele obținute sunt prezentate în Tabelul 4.4 pentru accelerometru, în Tabelul 4.5 pentru giroscop și în Tabelul 4.6 pentru magnetometru.

Tabelul 4.4 Influența domeniului de măsurare și a lățimii de bandă asupra entropiei accelerometrului

Domeniu de măsurare / Lățime de bandă	2 g	4 g	8 g
3330 Hz	0,6438	0,5133	0,4194
417 Hz	0,4957	0,3831	0,3382
53 Hz	0,3049	0,2549	0,1904

Tabelul 4.5 Influența domeniului de măsurare și a lățimii de bandă asupra entropiei giroscopului

Domeniu de măsurare / Lățime de bandă	245 dps	500 dps	1000 dps
1250 Hz	0,3931	0,4337	0,3356
312 Hz	0,3401	0,2386	0,2118
22 Hz	0,2094	0,1736	0,1083

Tabelul 4.6 Influența domeniului de măsurare asupra entropiei magnetometrului

Domeniu de măsurare	4 Gauss	8 Gauss	12 Gauss
	0,4414	0,4137	0,3850

Din rezultatele prezentate în Tabelul 4.5 și Tabelul 4.6 se poate observa evident că nivelul entropiei scade direct proporțional cu valoarea domeniului de măsurare și invers proporțional cu lățimea de bandă.

- *Frecvența de eșantionare*

Frecvența de eșantionare ar putea fi un element care să influențeze valoarea entropiei. Pentru a stabili acest aspect au fost achiziționate date de la senzori în cazul unor frecvențe de eșantionare diferite. Și în acest caz a fost izolată influența celeorlalți factori asupra valorilor entropiei și testele au fost efectuate doar cu o singură platformă în cazul în care nu am aplicat mișcare suplimentară asupra senzorilor. Rezultatele sunt prezentate în Tabelul 4.7 pentru accelometru și în Tabelul 4.8 pentru giroscop. Concluzia care se poate extrage din aceste rezultate este că frecvența de eșantionare nu influențează valoarea entropiei.

Tabelul 4.7 Influența frecvenței de eșantionare asupra entropiei accelerometrului

Frecvență de eșantionare/ Lățime de bandă	Accelerometru
6660 Hz / 1666 Hz	0,6212
3330 Hz / 1666 Hz	0,6142

Tabelul 4.8 Influența frecvenței de eșantionare asupra entropiei giroscopului

Frecvență de eșantionare/ Lățime de bandă	Giroscop
6660 Hz / 173 Hz	0,2278
3330 Hz / 172 Hz	0,2414

- *Platforma de achiziție*

Un alt element care ar putea influența entropia este platforma de achiziție. Chiar dacă se folosește același tip de platformă, anumite elemente din componența acesteia ar putea introduce zgomote suplimentare sau erori de măsurare, care pot influența într-un sens sau altul valoarea entropiei. Pentru a analiza acest aspect am comparat valorile entropiei extrase de la doi senzori identici MPU9250, conectați pe rând la trei plăci de achiziție de tip Arduino UNO și o platformă Arduino Atmega 256. Testele au fost efectuate pentru doi dintre senzori: accelerometrul și giroscopul. Rezultatele sunt prezentate în Tabelul 4.9 pentru accelometru și în Tabelul 4.10 pentru giroscop. Comparând valorile entropiei obținute în toate cazurile analizate, se poate observa faptul că, în cazul accelerometrului, valorile entropiei sunt foarte apropiate, iar în cazul giroscopului sunt mici variații dar nesemnificative.

Tabelul 4.9 Influența platformei de achiziție asupra entropiei accelerometrului

Placa de achiziție	Senzor MPU9250 - 1	Senzor MPU9250 - 2
Arduino UNO – placa 1	0,4618	0,4626
Arduino UNO – placa 2	0,4660	0,4618
Arduino UNO – placa 3	0,4671	0,4671
Atmega 256	0,4670	0,4571

Tabelul 4.10 Influența platformei de achiziție asupra entropiei giroscopului

Placa de achiziție	Senzor MPU9250 - 1	Senzor MPU9250 - 2
Arduino UNO – placa 1	0,2690	0,2708
Arduino UNO – placa 2	0,2371	0,2400
Arduino UNO – placa 3	0,2513	0,2586
Arduino Atmega 256	0,2258	0,2387

4.2.2.2. Analiza stabilității sursei de entropie

- *Analiză pe termen lung*

Mai importantă decât valoarea entropiei pe care o poate genera o sursă este capacitatea de a menține în timp acest nivel de entropie. În acest scop este necesară o analiză pe termen lung a nivelului de entropie generată și a numărului de eșecuri rezultate în cazul aplicării unor

teste de sănătate. De asemenea, comportamentul sursei în diferite moduri de funcționare poate oferi informații referitoare la modalitatea de implementare și utilizare a sursei de entropie.

În Tabelul 4.11 pentru accelorometru, în Tabelul 4.12 pentru giroscop și în Tabelul 4.13 pentru magnetometru sunt prezentate valorile minime, maxime și medii ale entropiei calculate pentru fiecare caz în parte. De asemenea, este calculată și deviația standard, pentru a pune în evidență variațiile valorilor entropiei pe un timp îndelungat. Din valorile prezentate se poate observa că valorile entropiei, în toate cazurile analizate, nu variază foarte mult, având o deviație standard de ordinul 10^{-2}.

Tabelul 4.11 Analiza entropiei pe termen lung în cazul accelerometrului

Parametru analiza stabilitate / Platformă/tip mișcare	**Valoare medie entropie**	**Valoare minimă entropie**	**Valoare maximă entropie**	**Deviație standard**	**Număr de seturi**
Platforma 1-fără mișcare	0,6499	0,6379	0,6638	0,0058	100
Platforma 1- autoturism în mișcare	0,8480	0,8076	0,8878	0,0263	10
Platforma 2-fără mișcare	0,4732	0,4503	0,5086	0,0139	30

Tabelul 4.12 Analiza entropiei pe termen lung în cazul giroscopului

Parametru analiza stabilitate / Platformă/tip mișcare	**Valoare medie entropie**	**Valoare minimă entropie**	**Valoare maximă entropie**	**Deviație standard**	**Număr de seturi**
Platforma 1-fără mișcare	0,4154	0,3427	0,4619	0,0222	100
Platforma 1- autoturism în mișcare	0,4295	0,3564	0,5132	0,0475	10
Platforma 2-fără mișcare	0,2347	0,1840	0,2745	0,0287	30

Tabelul 4.13 Analiza entropiei pe termen lung în cazul magnetometrului

Parametru analiza stabilitate / Platformă/tip mișcare	**Valoare medie entropie**	**Valoare minimă entropie**	**Valoare maximă entropie**	**Deviație standard**	**Număr de seturi**
Platforma 1-fără mișcare	0,4411	0,4189	0,4658	0,0108	100
Platforma 1- autoturism în mișcare	0,5521	0,4964	0,5970	0,0268	10

- *Analiză la restartare*

Sursele de entropie pot funcționa diferit imediat după restartare în comparație cu funcționarea într-un regim normal.

Pentru a determina comportamentul sursei de entropie bazată pe senzori de mișcare la restart, am restartat-o de un număr 1.000 de ori, de fiecare dată fiind colectate un număr de 1.000 de secvențe de la fiecare senzor în parte. După fiecare etapă de colectare de date, sursa a fost oprită pentru un interval de 15 minute, timp necesar pentru a simula revenirea platformei într-o stare de repaus. Din datele colectate a fost creată o matrice de $M_{i,j}$ cu 1.000 de linii și 1.000 de coloane. Concatenând datele pe linii și apoi pe coloane se creează două seturi de date.

Pentru datele colectate, testele de sănătate au fost validate folosind librăria software din [16]. În Tabelul 4.14 sunt prezentate valorile entropiei obținute pentru cele două seturi de date. Se poate observa că valorile entropiei obținute la restart sunt mai mari decât cele obținute în regim normal de funcționare, putându-se astfel afirma că sursa de entropie nu este influențată în sens negativ de valorile colectate la restart.

Tabelul 4.14 Analiza la restartare

Tipul de senzor	Entropie în regim normal de lucru	Entropie la restart – set de date concatenat pe linii	Entropie la restart – set de date concatenat pe coloane
Accelerometru	0,63	0,90	0,88
Giroscop	0,34	0,47	0,50
Magnetometru	0,41	0,46	0,43

- *Teste de sănătate*

Soluția prezentată în această teză implementează două teste continue de sănătate prezentate în [15]: testul repetiției și testul proporțiilor adaptive.

Pentru a analiza stabilitatea sursei de entropie am estimat pe o perioadă îndelungată numărul de erori raportate de cele două teste de sănătate mentionate mai sus. Concluziile acestei analize sunt importante pentru a decide dacă soluția propusă este pretabilă pentru utilizarea în aplicații reale. În cazul în care numărul de erori ar fi foarte frecvent, disponibilitatea sursei ar fi redusă, la fel și viteza de generare. Pentru a susține această analiză, am numărat erorile raportate de testele de sănătate pentru diferite valori ale probabilității α. Pentru a obține o acuratețe cât mai bună a rezultatelor, experimentele au fost realizate folosind un număr mare de secvențe, și anume 100.000.000 pentru fiecare dintre cei trei senzori: accelerometru, giroscop și magnetometru. Rezultatele obținute sunt prezentate în Tabelul 4.15 pentru testul repetiției și în Tabelul 4.16 pentru testul proporțiilor adaptive.

Tabelul 4.15 Testul repetiției

Senzor / ***α***	**Accelerometru**		**Giroscop**		**Magnetometru**	
	C	**Număr erori**	**C**	**Număr erori**	**C**	**Număr erori**
$\mathbf{2^{-20}}$	4	10	8	0	7	0
$\mathbf{2^{-15}}$	3	1075	6	2	5	10

Tabelul 4.16 Testul proporțiilor adaptive

Senzor / *α*	Acelerometru		Giroscop		Magnetometru	
	C	**Număr erori**	**C**	**Număr erori**	**C**	**Număr erori**
2^{-20}	38	0	120	0	89	0
2^{-10}	30	0	105	0	76	0
2^{-5}	24	0	94	0	67	0

Din analiza rezultatelor prezentate în tabelele de mai sus se poate observa că nu sunt raportate erori, cu o singură excepție. Doar în cazul accelerometrului un număr de zece erori au fost raportate pe perioada analizată pentru testul repetiției, ceea ce înseamnă că sursa de entropie ar putea genera o eroare la un interval de 785 de zile.

4.2.2.3. Analiza rezistenței la atacuri

- *Atacuri pasive*

Pentru a analiza rezistența soluției prezentate în cazul unui atac pe canal alăturat, am realizat un experiment prin care am colectat date simultan cu două platforme de achiziție identice. În cadrul experimentului am încercat, pe cât posibil, ca achiziția de date să se realizeze în aceleași condiții pentru cele două platforme. Secvențele extrase cu cele două platforme au fost analizate folosind două instrumente matematice utilizate pentru compararea unor șiruri de date: coeficientul de corelație Pearson și distanța Hamming.

Datele au fost extrase pentru fiecare axă în parte, pentru toți cei trei senzori de mișcare: accelerometru, giroscop și magnetometru. Cu scopul de a analiza dacă corelația depinde de nivelul de entropie generat de sursă, am colectat date de la senzori în mai multe situații. Nivelul de entropie diferit a fost obținut prin parametrizarea senzorilor.

În Tabelul 4.17 sunt prezentate valorile coeficientului Pearson calculate pentru diferite valori ale entropiei generate de accelorometru, giroscop și magnetometru. Analiza s-a efectuat pe siruri are contin 512 biti de entropie, dublu decat dimensiune unei chei simetrice. În functie de entropia generată de senzor șirul analizat are o dimensiuni diferite (Ni). După cum se poate observa din valorile prezente în tabel, corelația dintre cele două șiruri este aproape inexistentă. De asemenea, aceasta nu depinde de valoarea entropiei sau a lungimii șirului de date.

Tabelul 4.17 Analiza corelației folosind coeficientul Pearson

Senzor	**Acelerometru**			**Giroscop**			**Magnetometru**		
Entropie	0,64	0,38	0,19	0,43	0,23	0,10	0,44	0,41	0,38
N_i	100	168	337	148	269	591	145	155	167
P - axa Y	0,0813	0,0695	0,0520	0,0626	0,0582	0,0807	0,0748	0,0812	0,0665
P - axa X	0,0744	0,0634	0,0509	0,0636	0,0607	0,0524	0,0603	0,0616	0,0760
P - axa Z	0,0836	0,0637	0,0649	0,0619	0,0593	0,0581	0,0682	0,0672	0,0733

În Tabelul 4.18 sunt prezentate valorile obținute pentru distanțele Hamming în toate cazurile prezentate mai sus. De asemenea, în tabel mai sunt prezentate și deviațiile medii ale acestor valori pentru fiecare senzor și nivel de entropie analizat. A fost aleasă prezentarea rezultatelor sub această formă deoarece poate oferi o reprezentare mai bună a corelației dintre cele două șiruri analizate. Formula de calcul pentru deviația medie este prezentată în (4.1).

$$\mathrm{DH} = \frac{|DHx-4| + |DHy-4| + |DHz-4|}{3} \tag{4.1}$$

Tabelul 4.18 Analiza corelației folosind distanța Hamming

Senzor	**Acelerometru**			**Giroscop**			**Magnetometru**		
Entropie	0,64	0,38	0,19	0,43	0,23	0,10	0,44	0,41	0,38
Deviația medie	0,03	0,38	0,73	0,02	0,84	1,54	0,13	0,34	0,47
P - axa Y	4,05	4,13	4,98	3,96	3,22	2,44	4,04	3,74	5,10
P - axa X	4,01	5,02	3,42	3,98	3,39	2,28	4,26	3,36	4,10
P - axa Z	3,95	3,99	3,36	4,03	2,84	2,66	3,90	3,86	3,77

Din analiza datelor din Tabelul 4.18 se poate observa că valorile distanței Hamming se află în jurul valorii de 4, ceea ce înseamnă că datele nu sunt corelate. Cu toate acestea, se poate observa din analiza valorilor deviației medii că gradul de corelație crește odată cu scăderea nivelului de entropie.

- *Atacuri active*

În cazul surselor de entropie bazate pe datele colectate de la senzori, atacurile pasive se realizează pentru a reduce nivelul de entropie specific funcționării sursei în condiții normale. În cadrul acestui studiu am analizat rezistența sursei la patru tipuri de atacuri:

- mișcarea de rotație repetitivă asupra senzorilor a fost analizat în cadrul analizei sursei de zgomot. Rezultatele au arătat că valoarea entropiei a crescut datorită faptului că s-a aplicat mai multă mișcare senzorilor. Concluzia ar fi că este aproape imposibil de a reproduce o mișcare repetitivă perfectă astfel încât să determine generarea de valori identice;

- saturarea senzorilor -în acest mod valorile de ieșire al senzorilor ar putea fi maxime, reducând valoarea entropiei la zero. Pentru a detecta astfel de atacuri, soluția propusă are implementate teste de sănătate capabile să identifice rapid astfel de comportamente anormale.
- frecvenței de eșantionare ar putea influența valorile entropiei generate de sursă. Acest aspect a fost analizat în cadrul analizei sursei de zgomot. Rezultatele prezentate în Tabelul 4.7 și Tabelul 4.8 infirmă influența negativă a valorii frecvenței de eșantionare asupra valorilor entropiei;
- modificarea temperaturii mediului în conjurător. Pentru a analiza influența acestui atac asupra valorilor entropiei am colectat date de la platforma de senzori în timp ce aceasta funcționa la temperaturi extreme cuprinse între -18ºC și + 82ºC. Valorile estimate ale entropiei în aceste situații le-am comparat cu valorile obținute la o temperatură uzuală de funcționare de +23ºC. Analizând rezultatele prezentate în Tabelul 4.19, se poate concluziona că în cazul unor temperaturi scăzute nivelul entropiei nu este afectat negativ iar în cazul unor temperaturi ridicate entropia colectată este mai mare decât în regim normal de lucru.

Tabelul 4.19 Analiza influenței temperaturii asupra entropiei

Temperatura / **Tipul de senzor**	**-18ºC**	**+23ºC**	**+82ºC**
Accelerometru	0,6313	0,6379	0,7596
Giroscop	0,4393	0,4153	0,4637
Magnetometru	0,4500	0,4411	0,6365

4.2.2.4. Analiza performanței sursei de entropie

- *Viteza de generare*

Calculul vitezei de generare a sursei de entropie a fost realizat pentru fiecare dintre cei trei senzori analizați: accelerometru, giroscop și magnetometru. Timpii de execuție au fost estimați cu ceasul intern al microcontrolerului.

În calculul vitezei de generare s-a ținut cont de următoarele aspecte:

- sursa de entropie generează secvențe de 256 de biți, care conțin 256 biți de entropie. În acest sens, de la senzor trebuie achiziționat un număr de secvențe suficient de mare astfel încât să conțină 512 de biți de entropie (de două ori mai mare decât ieșirea sursei, conform recomandărilor NIST din [15]);

- timpul de execuție considerat reprezintă timpul necesar pentru extragerea câte unei secvențe pentru fiecare axă a senzorului: axa x, axa y și axa z, însumat cu timpul necesar pentru execuția funcției SHA256.

În Tabelul 4.20 sunt prezentate vitezele de generare obținute pentru cei trei senzori împreună cu timpii de extragere pentru secvențe și valorile entropiei folosite în calcule.

Tabelul 4.20 Viteza de generare a sursei de entropie

Tipul de senzor	Accelerometru	Giroscop	Magnetometru
Timp extragere secvență (µs)	2560	2560	1540
Entropie secvență per bit	0,63	0,34	0,42
Viteza de generare (Kb/s)	2,88	1,56	3,12

După cum se poate observa din rezultatele prezentate în Tabelul 4.20, viteza cea mai mare s-a obținut pentru magnetometru, deși accelerometrul este cel care generează cea mai multă entropie per bit. Acest lucru se datorează faptului că timpul de extracție pentru magnetometru este mai mic decât cel pentru accelerometru.

- *Analiza de consum*

Analiza consumului de curent a sursei de entropie a fost realizată pentru fiecare dintre cei trei zenzori analizați: accelerometru, giroscop și magnetometru. Timpii de execuție au fost estimați cu ceasul intern al microcontrolerului.

În analiza de consum s-a ținut cont de următoarele aspecte:

- sursa de entropie generează secvențe de 256 de biți care conțin 256 biți de entropie, ceea ce înseamnă că numărul de secvențe utilizate este dependentă de valoarea entropiei.

- valoarea curentului consumat de platformă ia în considerare curentul consumat de sensor I_{DD_S} (conform informațiilor din datasheet), curentul consumat de microcontroller - $I_{DD\_MCU\_S}$ pe timpul colectării datelor de la sensor și curentul consumat de microcontroller pentru calculul funcției SHA256 (măsurați pe pinul JP5 al microcontrolerului);

În Tabelul 4.21 sunt prezentate următoarele informații: numărul de biți care pot fi generați având la dispoziție o baterie de 1000 mAh, valorile curenților și valorile entropiei folosite în calcule.

Tabelul 4.21 Analiza de consum pentru sursa de entropie

Tipul de senzor	Accelerometru	Giroscop	Magnetometru
I_{DD_S} (mA)	13,81	13,81	13,81
$I_{DD\_MCU\_S}$ (mA)	0,16	0,49	0,27
Entropie secvență per bit	0.63	0,34	0,42
Numărul de biți generați (Mb)	485,07	303,30	508,30

După cum se poate observa din rezultatele prezentate în Tabelul 4.21, cel mai mic current consumat îl are soluția care folosește magnetometrul, fiind posibilă generarea celui mai mare număr de biți folosind curentul dintr-o baterie cu capacitate de 1000 mAh. Valorile

prezentate în tabel sunt orientative, deoarece în aplicații reale platformele de achiziții conțin și alte componente pe lângă microcontroler care consumă curent.

5. Generator de numere aleatoare pentru utilizare în aplicații IoT

În acest capitol propun un generator de numere aleatoare sigur pentru utilizare pe dispozitive IoT cu resurse limitate. După analiza provocărilor implementării generatoarelor de numere aleatoare în aplicații IoT și nivelurilor de securitate pe care acestea le pot atinge, se propune o soluție care folosește resurse limitate și asigură cel mai puternic nivel de securitate. Soluția propusă este optimizată prin realizarea unor experimente prin intermediul cărora se identifică o soluție optimă pentru sursa de zgomot și algoritmul determinist. În final se realizează o analiză de securitate și de eficiență, realizându-se o comparație cu o soluție clasică bazată pe algoritmul AES în modul CTR.

5.1. Soluția propusă

Generatorul de numere aleatoare propus este proiectat pentru a fi implementat pe dispozitive IoT. Soluția care fi prezentată în continuare are în vedere cerințele referitoare la resurse limitate specifice dispozitivelor IoT, asigurând în același timp cel mai înalt grad de securitate impus generatoarelor de numere aleatoare. Schema generală a soluției de generator de numere aleatoare este prezentată în Figura 5.1. Pentru a asigura proprietățile de securitate, folosind în același timp resurse limitate, am selectat un algoritm lightweight implementat într-o schemă de tip AEAD (în engleză Authenticated Encryption with Associated Data). Această schemă folosește un mesaj (text clar) de lungime 128 de biți, pe care îl criptează folosind un algoritm de criptare. Cheia de criptare utilizată de algoritm este regenerată pentru fiecare apel al funcției de generare numere aleatoare. Deoarece aceasta reprezintă ieșirea unei surse de entropie, valoarea acesteia nu poate fi estimată. La fiecare iterație a algoritmului se utilizează un nonce – un număr generat aleator la inițializarea generatorului folosind sursa de entropie, care se incrementează la fiecare iterație.

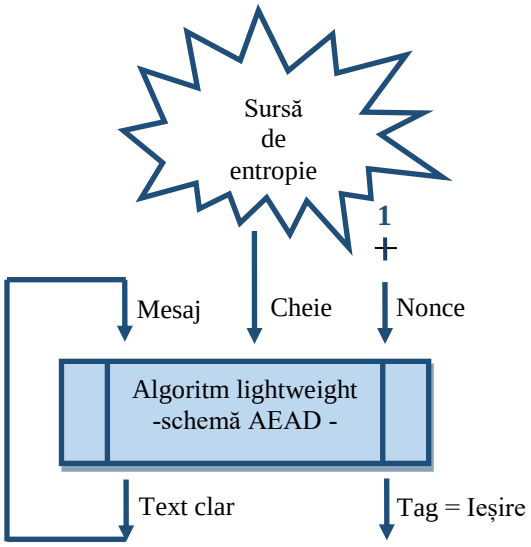

Figura 5.1 Schema de implementare a generatorului de numere aleatoare

În faza de inițializare a generatorului toate intrările acestuia sunt generate folosind date preluate de sursa de entropie. Ieșirile generatorului sunt blocuri de date lungime 128 de biți. Pentru a genera secvențe de date mai mari algoritmul rulează în buclă, în sensul că se utilizează textul criptat pentru mesaj – textul clar în următoarea iterație. Datele de ieșire sunt reprezentate de valoarea tag-ului, ce reprezintă un cod de autentificare a mesajului. Această schemă de utilizare a datelor de ieșire are avantajul că permite generarea de date de lungime variabilă. Faptul că s-a ales utilizarea tag-ului pentru datele de ieșire are avantajul că parametrii interni ai algoritmului, precum mesajul – testul clar și textul criptat, nu sunt expuși la ieșirea generatorului pentru a fi utilizați de un eventual atacator.

Sursa de entropie folosită își extrage aleatorismul de la datele provenite de la senzori de mișcare. Această soluție a fost aleasă deoarece este foarte ușor de implementat în multe aplicații IoT, care folosesc astfel de senzori, nefiind necesar hardware suplimentar pentru realizarea acesteia. Pentru implementarea sursei de entropie s-a folosit modulul multi-chip MPU 9250, care conține trei tipuri de senzori în trei axe de tip MEMS (Micro Electro-Mechanical Systems): accelerometru, giroscop și magnetometru.

Datele extrase de la senzori sunt digitizate pe 16 biți, însă nu toți acești biți sunt purtători de entropie. Din analiza pe care am realizat-o în studiul din [8] pe senzori similari, având aceleași setări, datele fiind colectate când acesția nu se aflau în mișcare, se poate observa (Figura 5.2 pentru acceleromentru și Figura 5.3 pentru giroscop) că doar cei mai puțin semnificativi biți de pe fiecare axă pot genera entropie.

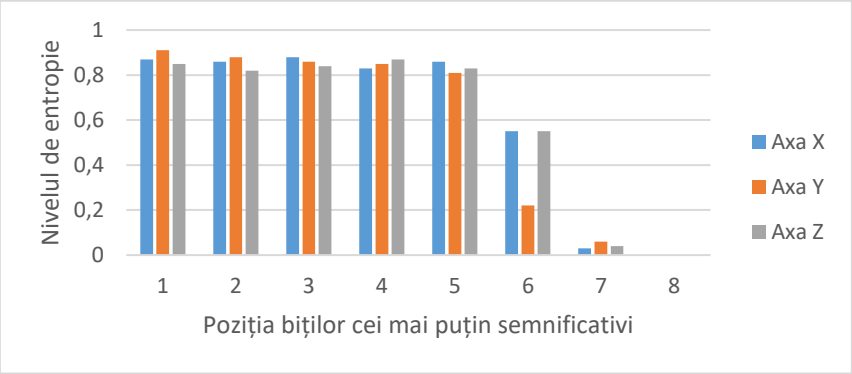

Figura 5.2 Entropia estimată pentru biții de pe fiecare axă pentru accelerometru

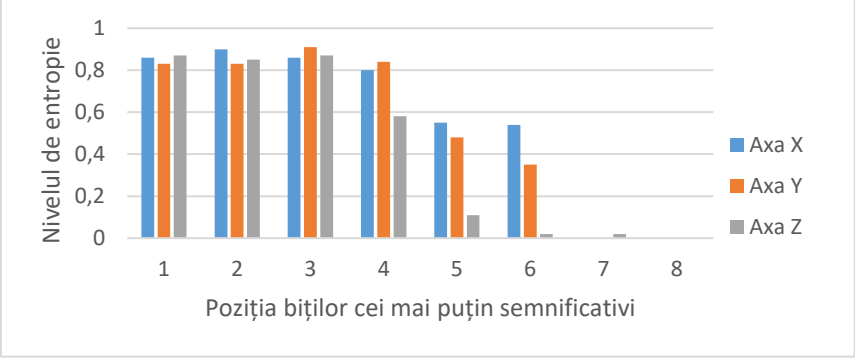

Figura 5.3 Entropia estimată pentru biții de pe fiecare axă pentru giroscop

Având în vedere aceste aspecte, soluția propusă pentru sursa de entropie extrage cei mai puțin semnificativi 4 biți de la fiecare axă a senzorului, pe care îi concatenează pentru a crea secvențe de 128 de biți necesare pentru intrările DRBG-ului.

5.2. Analiza și evaluarea soluției propuse

5.2.1. Analiza de securitate

Pentru a realiza analiza de securitate a soluției propuse am luat în considerare cerințele propuse de Oficiul federal al securității informațiilor german în metodologia de evaluare a generatoarelor de numere aleatoare AIS 20/ AIS 30, publicată inițial în 2011 în [17] și actualizată în 2022 în [18]. Această metodologie vine în sprijinul realizării unei evaluări de securitate a generatoarelor de numere aleatoare folosind Criteriile Comune.

Soluția propusă în acest studiu respectă cerințele funcționale ale clasei PTG.3 conform recomandărilor AIS 20/AIS 30. Această clasă recomandă cea mai sigură schema de generator de numere aleatoare, având asigurate următoarele proprietăți de securitate:

- Backward secrecy;
- Forward secrecy;
- Enhanced backward secrecy;
- Enhanced forward secrecy;

Ultimul aspect, dar cel mai important, se referă la aleatorismul datelor generate. Evaluarea generatorului de numere aleatoare am realizat-o folosind bateria de teste statistice NIST_STS. Testele au fost efectuate pe secvențe de date de 131,072 octeți. Întrucât o evaluare din punct de vedere statistic este cu atât mai exactă cu cât volumul de date testat este mai mare, am aplicat testele pentru un număr de 1.000 de secvențe distincte pentru fiecare caz analizat.

Rezultatele testării sunt prezentate în Tabelul 5 1. Analizând rezultatele obținute, se poate observa că pentru toate cazurile, doar unul sau două teste sau subteste nu au trecut. Având în vedere acest aspect, cât și faptul că toate testele relevante au trecut, se poate considera că toate soluțiile analizate prezintă proprietăți de aleatorism foarte bune.

Tabelul 5 1 Rezultatele testării statistice

Tip RNG	Dimensiune ieșire date generate	Teste trecute	Teste picate
RNG_Comet	128 biți	186 din 187	The Non-overlapping Template Matching Test – 1 subtest
	4096 biți	187 din 187	
	1 Mb	187 din 187	
RNG_Sparkle	128 biți	186 din 187	The Random Excursions Variant Test - 1 subtest
	4096 biți	187 din 187	
	1 Mb	185 din 187	The Non-overlapping Template Matching Test – 2 subteste
RNG_Romulus	128 biți	185 din 187	The Non-overlapping Template Matching Test - 2 subteste
	4096 biți	185 din 187	The Non-overlapping Template Matching Test - 2 subteste
	1 Mb	183 din 187	The Discrete Fourier Transform (Spectral) Test The Non-overlapping Template Matching Test – 3 subteste
RNG_Photon	128 biți	187 din 187	
	4096 biți	186 din 187	The Random Excursions Variant - Test 1 subtest
	1 Mb	186 din 187	The Discrete Fourier Transform (Spectral) Test

5.2.2. Analiza eficienței

Pentru a identifica soluția care oferă cea mai bună eficiență din punctul de vedere al puterii consumate și al resurselor necesare pentru realizarea implementării am efectuat o serie de experimente. Elementele pe care le-am luat în considerare în analiză, au fost: eficientizarea colectării datelor pentru sursa de entropie, identificarea unui algoritm criptografic pentru DRNG care să necesite cele mai puține resurse și implementarea soluției pe o platformă care poate fi utilizată și în aplicații IoT.

Pentru a identifica varianta optimă pentru sursa de entropie am analizat entropia generată de accelerometru și giroscop. Pentru a obține cel mai mare nivel de entropie în situația în care senzorii nu sunt în mișcare am analizat în lucrarea [7] posibilitatea de parametrizare a senzorilor. Astfel, am aflat că lățimea de bandă și domeniul de măsurare pot influența nivelul entropiei generată de senzori. În urma testelor efectuate a reieșit că, setând domeniul de măsurare la valoarea cea mai mică (2g pentru accelerometru și 245 dps pentru giroscop) și lățimea de bandă la valoarea cea mai mare (3330 Hz pentru accelerometru și 1250 Hz pentru giroscop), se obțin cele mai mari valori pentru entropie.

Întrucât am observat că timpii de extracție a datelor de la senzori sunt diferiți, am încercat diferite combinații pentru cei doi senzori. În Tabelul 5.2 sunt prezentați timpii necesari pentru extracția datelor în aceste cazuri, alături de nivelul de entropie al datelor extrase..

Tabelul 5.2 Viteza de generare a sursei de entropie

Tip senzor	Axa	Entropie	Timp extracție 1 bit de entropie (ns)
Giroscop	X	4,35	204,60
Giroscop	Y	4,51	197,34
Giroscop	Z	4,43	200,90
Giroscop	*XYZ*	*4,48*	*96,73*
Accelerometru	X	2,71	697,42
Accelerometru	Y	3,02	625,83
Accelerometru	Z	2,92	647,26
Accelerometru	XYZ	3,3	232,32
Accelerometru / Giroscop	XYZ	3,4	157,84

Următoarea optimizare a avut în vedere identificarea unei variante optime de algoritm lightweight pentru implementarea DRNG-ului. Pentru implemetarea algoritmilor s-a utilizat o placă de tip Arduino Mini. Testele au fost efectuate pentru diferiți algoritmi lightweight.. Pentru a demonstra faptul că soluția propusă aduce îmbunătățiri de performanță, am comparat rezultatele obținute cu soluția AES_CTR propusă de NIST în [6].

În primul rând am analizat resursele necesare pentru implementarea soluției. În Tabelul 5.3 sunt prezentate, pentru fiecare implementare, spațiul de stocare pentru codul sursă și memoria dinamică necesară rulării acestuia. Valorile sunt prezentate în număr de octeți și ca procent din memoria totală disponibilă. Analizând rezultatele se poate observa că soluțiile prezentate necesită o capacitate de memorie comparabilă cu varianta AES_CTR, excepție făcând algoritmul Romulus. În schimb, AES CTR necesită cam cu 30% mai multă memorie dinamică.

Tabelul 5.3 Resurse necesare implementării RNG

Tip RNG	**Spațiu stocate program**		**Memorie dinamică**	
	Număr octeți	**Procent**	**Număr octeți**	**Procent**
RNG_Photon	9322	28%	601	29%
RNG _Sparkle	7418	22%	523	25%
RNG _Romulus	15584	48%	523	25%
RNG _Comet	8226	25%	523	25%
RNG_AES_CTR	8494	26%	841	41%

În al doilea rând, am analizat vitezele de generare a numerelor aleatoare pentru toate cele cinci implementări. Testele au fost efectuate pentru generarea celor mai uzuale dimensiuni de chei (128, 256, 512 , 1024, 2048 , 4096 de biți) dar și a unor secvențe mai lungi de 1Kb și 10 Kb. Analizând rezultatele prezentate în Figura 5.4, se poate observa că doar soluțiile implementate cu algoritmii Comet și Sparkle reușesc să obțină viteze de generare mai bune decât varianta AES_CTR.

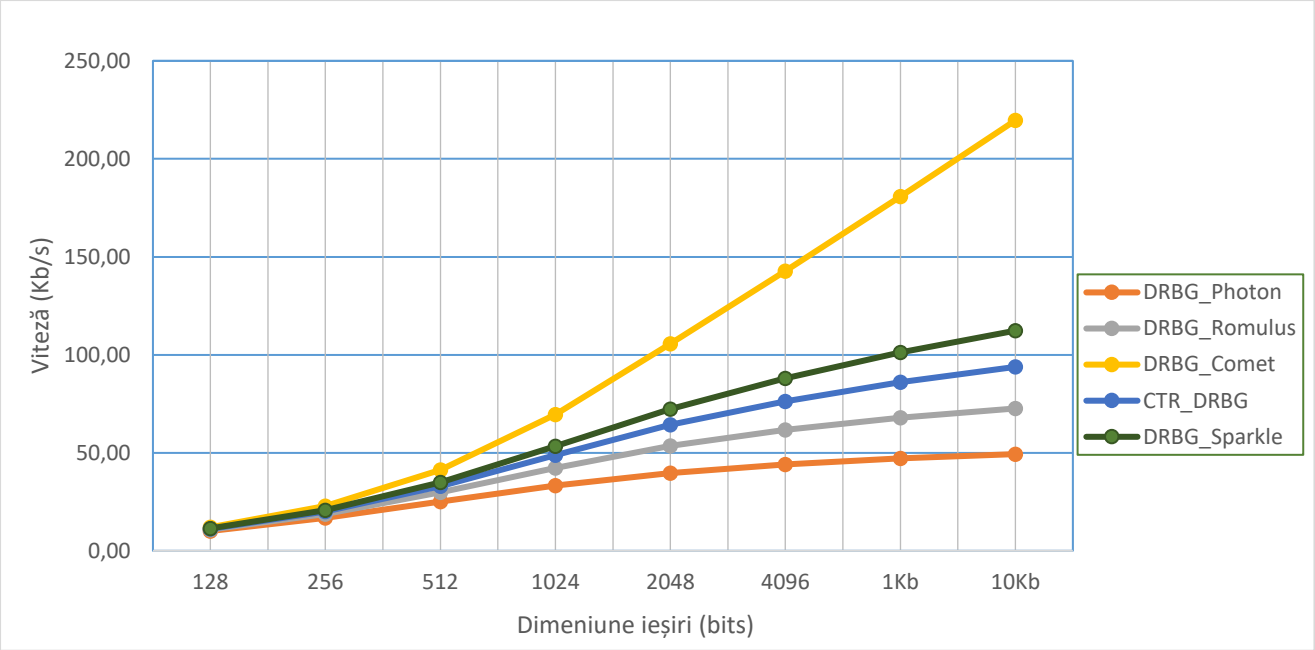

Figura 5.4 Vitezele de generare RNG-urilor

De asemenea am analizat variantele de DRNG din punct de vedere al necesarului de putere. Măsurătorile au fost realizate, folosind multimetrul professional FLUKE 8864A pentru a măsura curentul pe alimentarea plăcii Arduino Mini în timp ce pe aceasta rula programul care genera secvențe de 128 de biți, reinițializarea fiind realizată la fiecare iterație. Deoarece senzorul MPU9250 este alimentat de pe placa Arduino Mini, curentul măsurat reprezintă valoarea totală necesară pentru rularea RNG-ului. Având la dispoziție această valoare, se poate estima cu acuratețe capacitatea bateriei necesară pentru alimentarea acestei soluții pentru generarea unei anumite cantități de date aleatoare.

În Tabelul 5.4 sunt prezentate puterile consumate de cele cinci soluții și numărul de chei generate cu o baterie cu o capacitate de 1000 mAh. Puterea a fost calculată înmulțind curentul consumat de placa Arduino Mini pe care am realizat implemetările cu tensiunea acesteia de alimentare de 5V. Valoarea curentului reprezintă o medie a valorilor înregistrate pe o perioadă de 10s. Consumul de curent al microcontrolerului măsurat în stare de standby este 0,257 mA, mult mai mic decât în stare de funcționare. Astfel, se poate afirma că numărul de chei generat cu o baterie de 1000 mAh se poate obține și în condiții reale.

Tabelul 5.4 Analiza de consum pentru soluțiile RNG

Tip RNG	Putere consumată (mW)	Număr chei generate cu o baterei de 1000 mAh
RNG_Sparkle	87,0	$18.91*10^6$
RNG_Comet	86,5	$19.98*10^6$
RNG_Photon	88,0	$16.53*10^6$
RNG_Romulus	88,5	$17.59*10^6$
AES_CTR	86,5	$18.62*10^6$

Analizând valorile din Tabelul 5.4 se poate observa că cele mai bune performanțe se pot obține folosind algoritmul Comet, fiind urmat de Sparkle și soluția AES_CTR.

6. Concluzii

Analiza contextului de securitate actual în infrastructurile IoT indică faptul că mai sunt suficiente probleme de rezolvat în acest domeniu. În teză sunt prezentate câteva soluții care aduc îmbunătățiri pentru asigurarea securității datelor vehiculate pe dispozitive IoT.

Teza include contribuții originale teoretice și practice în domeniul asigurării securității datelor în medii restrictive IoT. Astfel, sunt propuse soluții arhitecturale, de integrare a unor tehnologii diferite, metodologii de analiză, dar și analize de securitate sau eficiență. Principalele contribuții descrise în detaliu sunt prezentate în continuare:

- Soluție arhitecturală de integrare a tehnologiei blockchain într-o infrastructură IoT de tip fog computing. În urma realizării unei analize pentru identificarea tipului de arhitectură IoT pentru care implementarea funcționalităților BC este realizabilă, am propus soluția optimă;
- Soluții arhitecturale de implementare a funcționalităților unui nod de senzori IoT și ale unui nod BC pe platforme FPGA. Am propus două soluții, luând în considerare consumul de putere. Am implementat componente esențiale pentru asigurarea funcționalităților unui nod de senzori și a unui nod BC pe platforme FPGA cu resurse diferite;
- Protocol simplu, eficient și sigur de stabilire a cheilor de sesiune pentru implementare pe noduri IoT. Protocolul a fost implementat pe o platformă utilizată în IoT, dotată cu un microcontroler, realizându-se o analiză din punctul de vedere al puterii consumate și a vitezei de execuție. Soluția folosește platforma blockchain Ethereum ca sursă de încredere, folosind un contract intelligent în acest scop. Soluția propusă a fost evaluată comparativ din punctul de vedere al eficienței și al costurilor cu soluția clasică protocol TLS – PKI;
- Soluție eficientă de sursă de entropie cu date extrase de la senzori. Soluția a fost optimizată prin identificarea parametrilor senzorilor pentru generarea unui nivel maxim de entropie în condiții diferite de utilizare. Nivelul de entropie generat a fost

evaluat folosind metrici standardizare NIST. Am realizat o analiză a performanței sursei din punctul de vedere al consumului de putere și al vitezei de generare;

- Metodologie originală de analiză a sursei de zgomot care colectează date de la senzori de mișcare. Analiza a fost efectuată având în vedere elementele componente ale sursei: fenomen fizic, senzor și platformă de achiziții. În acest scop, am realizat o serie de experimente care evidențiază influența fiecărui element în parte asupra valorii entropiei;
- Metodologie de analiză stabilitate a sursei de entropie. Analiza a fost efectuată pe două paliere. Am realizat o analiză pe termen lung pentru a identifica comportamentul în timp al sursei și o analiză a nivelului de entropie emis imediat după restartarea sursei;
- Metodologie de analiză a rezistenței la atacuri a unei surse de entropie care colectează date de la senzori. Am detaliat și realizat două tipuri de atacuri. Primul tip de atac a fost pasiv. Acesta a fost realizat pe canal alăturat, încercând estimarea datelor folosind o platformă de achiziție identică. A doilea tip de atacuri au fost active. Am construit o serie de atacuri pentru a detecta comportamentul sursei în cazul unor mișcări ciclice, în cazul saturației, în cazul funcționării la temperaturi extreme și în cazul modificării frecvenței de eșantionare;
- Soluție originală, sigură și eficientă de generator de numere aleatoare pentru utilizare în medii restrictive IoT. Implementarea soluției a fost realizată pe un microcontroler cu resurse limitate. Am efectuat o analiză din punctul de vedere al resurselor consumate, al consumului de putere și al securității, care cuprinde estimarea aleatorismului datelor generate, proprietăților de securitate și a tăriei criptografice a componentei deterministe.

Bibliografie

[1] M. Hogan, B. Piccaretta, "NISTIR 8200 - Interagency Report on the Status of International Cybersecurity Standardization for the Internet of Things (IoT)", November 2018

[2] https://ro.wikipedia.org/wiki/Internet, accesat la 16.03.2021

[3] https://www.statista.com/statistics/1194682/iot-connected-devices-vertically/ accesed on 16.03.2022

[4] *2018 Internet Security Threat Report*, Symantec Corporation, March 2018.

[5] A. Boteanu, F. Răstoceanu, I. Rădoi, C. Rusea, " Modeling and simulation of electromagnetic shielding for IoT sensor nodes case", 2019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD), 10-12 October 2019

[6] E. Barker, J. Kelsey, "NIST Special Publication 800-90C (Second Draft)-Recommendation for Random Bit 5 Generator (RBG) Constructions", Aprilie 2016

[7] F. Rastoceanu, R. Rughinis, S.D. Ciocirlan, M. Enache, "Sensor-Based Entropy Source Analysis and Validation for Use in IoT Environments", Electronics , 10(10), 1173, 2021

[8] F. Rastoceanu, B.I. Ciubotaru, I. Radoi, C.V. Marian, "Extented Analysis Using NIST Methodology of Sensors Data Entropy, U.P.B. Sci. Bull., Series C, Vol. 83, Iss. 2, 2021

[9] X. Zhu, I. Badr, "A Survey on Blockchain-based Identity and Access Management Systems for Internet of Things", 2018 IEEE Confs on Internet of Things, Iulie 2018

[10] D. B. Rawat , V. Chaudhary, R. Doku, “Blockchain Technology: Emerging Applications and Use Cases for Secure and Trustworthy Smart Systems”, *J. Cybersecur. Priv. , 1*(1), 4-18, 2021

[11] Michaela Iorga, Larry Feldman, Robert Barton Michael, J. Martin, Nedim, Goren Charif Mahmoudi, NIST Special Publication 500-325 - Fog Computing Conceptual Model, March 2018, https://doi.org/10.6028/NIST.SP.500-325

[12] F. Rastoceanu and R. Rughinis, "Blockchain Solution for Securing Fog-Computing Communications in IoT Applications," 2022 14th International Conference on Communications (COMM), 2022, pp. 1-6, doi: 10.1109/COMM54429.2022.9817211

[13] https://www.clickssl.net/low-cost-rapidssl-certificate, accesat la 15.09.2023

[14] F. Rastoceanu, I Radoi, ”FPGA based architecture for securing IoT with blockchain ” 2*019 International Conference on Speech Technology and Human-Computer Dialogue (SpeD)*, 2019, pp

[15] M.S. Turan, E. Barker, J. Kelsey, K. McKay, “NIST Special Publication 800-90B-Recommendation for the Entropy Sources Used for Random Bit Generation”, Ianuarie 2018

[16] https://github.com/usnistgov/SP800-90B_EntropyAssessment, accesat la data de 05.10.2023.

[17] W. Killmann, W. Schindler, “A proposal for: Functionality classes for random number generators”, Septembrie 2011

[18] Matthias Peter, Werner Schindler, A Proposal for Functionality Classes for Random Number Generators Version 2.35 – DRAFT, September 2, 2022