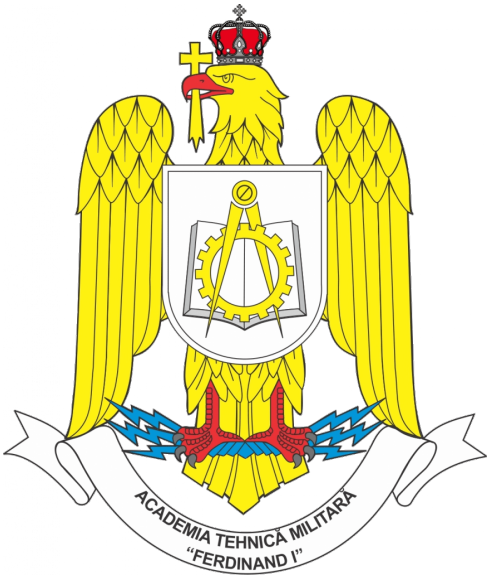
România
Ministerul Apărării Naționale
Academia Tehnică Militară "*Ferdinand I*"

Facultatea de Sisteme Informatice și Securitate Cibernetică
Calculatoare și Sisteme Informatice pentru Securitate si Apărare Națională



Unelte software bazate pe mecanisme de inteligență artificială aplicată în criptografie

Coordonator Științific
Cpt. Subașu Georgiana-Ramona

Absolvent
Std. Sg. Maj. Moldovan Andrei-Gabriel

București
2025

Abstract

Lucrarea propune o platformă integrată care combină diferiți agenți AI specializați ca utilitare de criptanaliză și audit de parole pentru a acoperi un flux complet de securitate modernă din punct de vedere criptografic. Sistemul folosește orchestrare prin microservicii containerizate, momentan, și include: un agent de decizie (SecureBERT) pentru extragere de intenție și de entități relevante; un detector de criptosisteme inspirat din CyberChef/dcode.fr; un orchestrator de spargere de hash-uri cu HashCat și John și generare de parole folosind PassGAN; un agregator de scoruri de parole (rețea neuronală, zxcvbn și HaveIBeenPwned), încă în lucru; un serviciu de testare primalitate/factorizare (YAFU + FactorDB); și un RAG local pentru asistență teoretică în criptografie. Se urmărește realizarea unei platforme centralizate care poate fi folosită local sau în cloud, cu capabilități AI/ML pentru detecție, analiză și suport educațional, punând accent pe modularitate, extindere ulterioară către Kubernetes și aliniere la bune practici de securitate.

Cuvinte cheie: inteligență artificială, criptanaliză, securitate cibernetică, parole, HashCat, John the Ripper, PassGAN, SecureBERT, RAG local, containerizare, microservicii, Kubernetes, FactorDB, YAFU, zxcvbn, HaveIBeenPwned.

Cuprins

Listă de Abrevieri

AI ..... artificial intelligence

ML ..... machine learning

PKI ..... public key infrastructure

RAG ..... retrieval-augmented generation

TLS ..... transport layer security

AES ..... Advanced Encryption Standard

DES/3DES .. Data Encryption Standard / Triple DES

RSA ..... Rivest–Shamir–Adleman (public-key cryptography)

ECC ..... elliptic curve cryptography

HMAC ..... hash-based message authentication code

AEAD ..... authenticated encryption with associated data

LLM ..... large language model

PQC ..... post-quantum cryptography

Capitolul 1: Introducere

1.1 Context general

Criptografia modernă a evoluat cu pași repezi, de la algoritmi simetrici clasici (DES, 3DES), către standarde robuste precum AES și suitele asimetrice (RSA, ECC), odată cu dezvoltarea infrastructurilor PKI și a protocoalelor TLS, precum și apariției PQC. În paralel, progresul AI a adus atât beneficii considerabile, cât și riscuri: modelele de tip ML/LLM pot accelera detecția anomaliilor și analiza traficului criptat, putând fi folosite și pentru generarea de atacuri (phishing, parole, malware) sau pentru asistarea criptanalizei. Astfel, securitatea cibernetică integrează din ce în ce mai mult tehnici AI precum răspuns rapid, corelare de evenimente și întărirea mecanismelor criptografice.

1.2 Problema abordată

Problema principală urmărește modul în care putem folosi agenți AI specializați pentru a automatiza analiza și răspunsul la amenințări criptografice, de la identificarea algoritmilor și auditul parolelor, până la factorizare pentru testare de securitate, cât și modul în care se poate îmbunătăți educația criptografică. Se observă, de asemenea, lipsa unei platforme integrate care să ofere aceste capabilități într-un mod modular, scalabil și sigur.

1.3 Scopul lucrării

Scopul lucrării este reprezentat de proiectarea unei platforme integrate, modulare și scalabile care folosește agenți AI pentru identificare de algoritmi, audit de parole, factorizare / teste criptografie, asistență teoretică, astfel încât detecția, analiza și răspunsul la amenințări să fie automatizate, reproductibile și ușor de folosit, atât local, cât și în cloud.

1.4 Obiective specifice

- Realizarea și testarea robustă a unei părți din agenții AI propuși.
- Integrarea tuturor agenților într-o infrastructură comună.
- Crearea unei soluții robuste de deploy și testare.
- Crearea unui Web Frontend și a unui Client CLI.
- Legarea componentelor într-un flux coerent.

Capitolul 2: Fundamente Teoretice

2.1 Elemente de bază ale criptografiei

Criptografia urmărește să asigure confidențialitate, integritate, autentificare și nerepudiere pentru date transmise pe canale nesigure. Se împart două familii majore: criptografia simetrică (aceeași cheie pentru criptare/decriptare) și criptografia asimetrică (pereche publică/privată), completate de funcții hash și coduri de autentificare a mesajelor (MAC/HMAC) [11, 9, 10]. Un criptosistem se definește formal prin spațiul mesajelor, al textelor cifrate, al cheilor și printr-o familie de funcții de criptare/decriptare parametrizate de cheie, iar securitatea se descrie prin modele de adversar și noțiuni riguroase de rezistență la atac. În practică, protocoalele moderne combină aceste primitive cu management de chei (generare, stocare, rotație) și cu mecanisme de transport securizat (ex. TLS), fiind fundamentele pe care se sprijină soluția propusă.

2.1.1 Definiții și clasificare

Din punct de vedere al clasificării, criptografia acoperă o suită vastă de primitive: criptare simetrică, criptare asimetrică, diverse criptosisteme (bloc, flux, hibrid), mecanisme de autentificare, funcții hash, semnături digitale, management al cheilor, generatoare pseudo-aleatoare și protocoale de distribuire a secretelor. Criptografia simetrică utilizează aceeași cheie pentru criptare și decriptare, oferind performanțe foarte bune și fiind preferată pentru volume mari de date, dar presupune existența unui canal sigur pentru distribuirea cheilor (exemplu: AES). Criptografia asimetrică (cu cheie publică) folosește o pereche publică/privată pentru a facilita schimbul securizat de chei, autentificarea și semnăturile digitale (RSA, ElGamal), însă costul computațional este mai ridicat.

Funcțiile hash criptografice produc un digest de lungime fixă dintr-un mesaj de lungime variabilă, fiind concepute să reziste la inversare și coliziuni, motiv pentru care sunt folosite la verificarea integrității și la stocarea sigură a parolelor. Alte primitive importante includ codurile de autentificare a mesajului (MAC/HMAC), care combină o cheie secretă cu o funcție hash sau cu o schemă de criptare pentru a valida autenticitatea și integritatea; scheme de semnătură digitală ce oferă non-repudiere fără partajarea cheii; generatoare de numere aleatoare criptografic sigure; și protocoale de negociere a cheilor, precum Diffie–Hellman sau ECDH. În practica modernă, sistemele combină aceste primitive: asimetria este folosită pentru a stabili secrete, simetria pentru traficul intens, hash-ul și MAC-urile pentru integritate și autentificare, iar managementul cheilor (generare, rotație, revocare) asigură durabilitatea modelului de securitate [7].

2.1.2 Standardizare

Organismele de standardizare în criptografie sunt esențiale pentru a asigura interoperabilitatea între sisteme, validarea algoritmilor criptografici și adaptarea tehnologiei la amenințările emergente, inclusiv cele cuantice. Exemple cheie includ NIST (FIPS 140 pentru module criptografice, FIPS 197/AES și seria FIPS 203–205 pentru algoritmi post-cuanți), ISO/IEC (ISO/IEC 19790 privind cerințele de securitate și ISO/IEC 18033 pentru algoritmi de criptare), ETSI (TS 104 015 dedicat criptografiei cuantice), ITU (familia ITU-T Y.3800 pentru distribuția cuantică de chei), BSI (TR-02102 cu recomandări de algoritmi) și IETF (RFC 5280 pentru PKI X.509). Ele stabilesc cerințele de certificare, mecanismele de testare și ghidurile de implementare care ghidează atât soluțiile civile, cât și cele militare [1].

2.2 Elemente fundamentale de inteligență artificială

Se folosesc mai multe concepte de IA aplicate:

- învățare supravegheată și clasificare semantică (modelul SecureBERT din componenta de decizie pentru extragerea intențiilor și entităților);
- embeddings obținute cu arhitecturi Transformer, folosite la căutare semantică în serviciul teoretic;
- RAG, care combină recuperarea de fragmente, reranking și un model LLM pentru a genera răspunsuri cu surse;
- reranking al rezultatelor de căutare pe baza scorurilor ML pentru a prioritiza contextul relevant;
- data augmentation în generatorul de întrebări (questions_generator) pentru a extinde seturile de antrenare cu variații și parafrazări sintetice;
- ensemble și agregare de scoruri (password checker combină rețea neuronală, zxcvbn și HIBP);
- inferență optimizată prin ONNX și containerizare pentru performanță și portabilitate;
- rolul LLM-ului în RAG: răspunsuri naturale, contextualizate, cu halucinații reduse prin ancorare în fragmentele recuperate.

2.3 Conexiunea AI–Criptografie

AI-ul este din ce în ce mai implicat în criptografie, atât ca instrument de analiză, cât și ca obiect al securizării. Modele AI specializate (ex. SecureBERT) pot automatiza identificarea algoritmilor și clasificarea cererilor tehnice, asistând în alegerea metodelor criptanalitice adecvate (brute-force sau dicționar), conform tiparelor datelor [4, 3]. Integrarea tehnicilor de Retrieval-Augmented Generation permite asistenților AI să furnizeze recomandări fundamentate pe literatura de specialitate și standarde (ex. FIPS 140-3, BSI TR-02102), susținând tranziția către algoritmi post-cuantici și reducând riscul de halucinații [2]. În evaluarea parolelor, se folosesc modele neuronale antrenate pe parole comune combinate cu estimatori euristici (zxcvbn) și baze de date publice precum HIBP,

ceea ce îmbunătățește precizia scorării și a analizei de risc [8, 12]. Generarea de scenarii și date sintetice prin modele generative facilitează testarea rezilienței sistemelor, iar sistemele de detecție bazate pe învățare profundă pot identifica comportamente suspecte în trafic sau log-uri [13]. Criptografia contribuie la protecția AI prin tehnici precum criptarea omomorfă, calculul multipartit securizat sau privacy diferențială, folosite pentru protejarea datelor și a modelelor în timpul antrenării sau inferenței [5]. Pentru integritatea modelelor, se propun watermark-uri digitale și semnături criptografice atașate acestora. Totuși, AI introduce riscuri semnificative, precum prompt injection sau exfiltrarea secretelor stocate, motiv pentru care sunt necesare guardrails, monitorizare și evaluare continuă a comportamentului modelului [6].

Capitolul 3: Soluția Propusă

3.1 Arhitectura generală

Descrie arhitectura macro a sistemului și include o diagramă care evidențiază principalele componente și fluxuri de date.

3.2 Metodologie și justificarea alegerilor

Explică abordarea metodologică, pașii parcurși și motivele pentru alegerea tehnologiilor și a modelelor AI.

3.2.1 Selecția tehnologiilor

Detaliază limbajele, cadrele și infrastructura utilizate și argumentează selecția lor.

3.2.2 Modele AI utilizate

Prezintă modelele AI integrate, modul în care sunt antrenate, configurate și orchestrate.

3.2.3 Principii de securitate

Descrie măsurile de securitate și mecanismele de protecție integrate încă din etapa de proiectare.

3.3 Descrierea componentelor

Analizează fiecare componentă a soluției folosind paragrafe dedicate ce includ rolul, funcționarea, diagrama și justificarea tehnologică.

3.3.1 Modul AI

Detaliază agentul AI, capabilitățile, datele utilizate și modul de interacțiune cu celelalte componente.

3.3.2 Modul criptografic

Descrie serviciile criptografice, protocoalele implementate și modul de integrare cu orchestrarea AI.

3.3.3 Orchestrator și interfețe

Prezintă modul de coordonare între componente, fluxurile de lucru și interfețele expuse (CLI, API, UI).

Capitolul 4: Implementare

4.1 Mediu de dezvoltare

Detaliază uneltele de dezvoltare, sistemul de operare, dependințele și configurațiile cheie.

4.2 Framework-uri și librării

Listează și descrie framework-urile, librăriile și modelele AI utilizate, inclusiv versiunile.

4.3 Containerizare și infrastructură

Prezintă modul în care aplicația este containerizată și orchestrată (Docker, docker-compose, Kubernetes etc.).

4.4 Măsuri de securitate aplicate

Enumeră controalele de securitate implementate (hardening, politici, rotație de chei, audit).

4.5 Automatizare

Descrie scripturile, pipeline-urile CI/CD sau orice automatizare care susține proiectul.

Capitolul 5:
Testare și Evaluare

5.1 Metodologia de testare

Descrie strategia generală de testare și criteriile utilizate pentru a valida soluția.

5.1.1 Teste unitare

Prezintă acoperirea testelor unitare, instrumentele folosite și exemple relevante.

5.1.2 Teste de integrare

Detaliază scenariile de integrare și modul în care componentele sunt validate împreună.

5.1.3 Teste de performanță

Specifică metodologiile și instrumentele folosite pentru măsurarea performanței și scalabilității.

5.2 Metrici și criterii de evaluare

Definește indicatorii cantitativi și calitativi utilizați în analizarea rezultatelor testelor.

5.3 Validarea rezultatelor

Analizează rezultatele obținute în raport cu obiectivele propuse și discută implicațiile.

Capitolul 6:
Rezultate și Discuții

6.1 Benchmark-uri comparative

Prezintă rezultate numerice și comparații cu soluții similare sau baseline-uri.

6.2 Analiza performanței

Evaluează latența, throughput-ul, utilizarea resurselor și scalabilitatea.

6.3 Beneficii și limitări

Rezumă beneficiile soluției și limitele identificate în urma testării.

6.4 Direcții de îmbunătățire

Propune optimizări și extensii viitoare.

Capitolul 7: Concluzii

7.1 Gradul de atingere a obiectivelor

Recapitulează modul în care au fost îndeplinite obiectivele definite în introducere.

7.2 Contribuții

Enumeră contribuțiile teoretice și practice ale lucrării.

7.3 Impact și dezvoltări viitoare

Evaluează impactul potențial și prezintă direcții de continuare a cercetării.

Bibliografie

[1] *Cryptographic Standardization*. `https://prism.sustainability-directory.com/term/cryptographic-standardization/`. Accesat 2025.

[2] Y. Dong et al. "ChatIoT: Large Language Model-based Security Assistant for Internet of Things with Retrieval-Augmented Generation". In: *arXiv preprint arXiv:2502.09896* (2025).

[3] H. Hu and K. Yuan. "Identification of Cryptographic Algorithms Based on CNN". In: *Proc. 4th Int. Conf. on Computer, Artificial Intelligence and Control (CAIC)*. 2025.

[4] B. D. Kim et al. "Cryptanalysis via Machine Learning Based Information Theoretic Metrics". In: *arXiv preprint arXiv:2501.15076* (2024).

[5] J.-W. Lee et al. "Privacy-Preserving Machine Learning with Fully Homomorphic Encryption for Deep Neural Network". In: *IEEE Access* 10 (2022), pp. 30039–30054.

[6] M. Q. Li and B. C. M. Fung. "Security Concerns for Large Language Models: A Survey". In: *arXiv preprint arXiv:2505.18889* (2025).

[7] Alfred J. Menezes, Paul C. Van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[8] J. Mo, H. Kuang, and X. Li. "Password Strength Detection via Machine Learning: Analysis, Modeling, and Evaluation". In: *arXiv preprint arXiv:2505.16439* (2025).

[9] Bruce Schneier. *Applied Cryptography*. Wiley, 1996.

[10] William Stallings. *Cryptography and Network Security*. Pearson, 2017.

[11] Douglas R. Stinson. *Cryptography: Theory and Practice*. CRC Press, 2005.

[12] Daniel L. Wheeler. "zxcvbn: Low-Budget Password Strength Estimation". In: *Proc. 25th USENIX Security Symposium*. 2016, pp. 157–173.

[13] Z. Xu et al. "Deep Learning-based Intrusion Detection Systems: A Survey". In: *arXiv preprint arXiv:2504.07839* (2025).

Capitolul 8:
Anexe

8.1 Listă de diagrame suplimentare

Include diagrame detaliate (de exemplu, PlantUML, diagrame de secvență sau arhitecturi alternative) care completează prezentarea din capitolele principale.

8.2 Configurații tehnice

Prezintă fișiere de configurare relevante (de exemplu, `docker-compose.yml`, fișiere de mediu, politici de securitate) și explică modul în care acestea sunt utilizate.

8.3 Scripturi și manual de utilizare

Include scripturile auxiliare și un ghid succint de utilizare a aplicației pentru diferitele roluri implicate.