

**Снігуров Аркадій Владиславович**

к.т.н., доцент,

декан факультету кібербезпеки, кафедри інфокомунікаційної інженерії ім. В.В. Поповського,
Харківський національний університет радіоелектроніки, Харків, Україна

ORCID ID 0000-0003-1355-7978

arkadii.snihurov@nure.ua

Заїка Павло Володимирович

студент факультету кібербезпеки,

кафедри інфокомунікаційної інженерії ім. В.В. Поповського,
Харківський національний університет радіоелектроніки, Харків, Україна

ORCID ID 0009-0000-4181-7429

pavlo.zaika@nure.ua

**МОДЕЛЬ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ ТА ЕФЕКТИВНОСТІ АТАК ПІДБОРУ
ПАРОЛІВ ІЗ ЗАЛУЧЕННЯМ OSINT ТА ВЕЛИКИХ МОВНИХ МОДЕЛЕЙ**

Анотація. Розглянуто сучасні вектори атак на паролі системи в умовах, коли обмеження кількості спроб знецінило повний перебір, а відкриті генеративні моделі та засоби розвідки за відкритими джерелами здешевили цільовий підбір, орієнтований на конкретного користувача. Показано, що наявні показники результативності (частка скомпрометованих облікових записів, час до першого влучення та теоретико-множинна складність) не зводяться до спільної шкали. Через це методи різної природи неможливо кількісно зіставити. Запропоновано модель обчислювальної складності, у якій вартість атаки подано як добуток позиції влучення у словнику на тривалість одного хешування. Модель конкретизовано для п'яти класів атак: повного перебору, корпусної, статистичної на основі ланцюгів Маркова, комбінаторної за OSINT-правилами та генеративної на основі великої мовної моделі. Введено параметр ефективності як відношення емпіричної ймовірності успіху до обчислювальної складності, доповнений оцінкою вартості невдалого прогону словника. Структура параметра відокремлює вплив словника, що діє через позицію влучення, від впливу алгоритму зберігання, що діє як спільний масштабувальний множник. Експериментальну перевірку виконано на еталонних паролях профілів із чотирма словниками та алгоритмами MD5, SHA-256, bcrypt і Argon2id. Комбінаторний OSINT-словник, у 57 разів менший за мільйонну вибірку RockYou, скомпрометував 66,7 % паролів. Корпусний і марковський словники не дали жодного точного збігу. Генеративний словник мав нижчу ймовірність успіху (3,3 %), але знайшов пароль на 232-й позиції. Його ефективність виявилася приблизно в 3,5 раза вищою, ніж у комбінаторного, і ця перевага не видна при порівнянні самих часток компрометації. Перехід до повільних алгоритмів знижує ефективність усіх атак приблизно на шість порядків без зміни їхнього відносного порядку. На основі результатів запропоновано пошарову систему заходів захисту, упорядковану за вимірним впливом на стійкість.

Ключові слова: паролі автентифікації; атаки підбору паролів; словникова атака; обчислювальна складність; позиція влучення; розвідка за відкритими джерелами (OSINT); великі мовні моделі; ланцюги Маркова; хешування

ВСТУП

Парольна автентифікація залишається переважним механізмом контролю доступу до інформаційних ресурсів, попри розвиток альтернативних методів ідентифікації. Водночас характер загроз для неї суттєво змінився. Обмеження кількості спроб та моніторинг підозрілої активності знецінили повний перебір, а поширення відкритих генеративних моделей і засобів розвідки за відкритими джерелами здешевило цільовий підбір, орієнтований на конкретного користувача. Класичне уявлення про стійкий пароль, засноване на його довжині та складі, втрачає силу, оскільки сучасні атаки експлуатують не формальну структуру пароля, а його зв'язок з особистими даними власника.



Оцінювання й зіставлення методів підбору ускладнені тим, що їхню результативність характеризують принципово різні величини. Частка скомпрометованих облікових записів описує ймовірність успіху, проте не враховує витрачених ресурсів. Час до першого влучення відображає вартість атаки, але залежить від апаратного забезпечення та параметрів алгоритму хешування [1]. Обчислювальна складність у теоретико-множинному вигляді характеризує простір пошуку, однак ігнорує вартість окремої перевірки, яка для сучасних повільних алгоритмів стає визначальною. Унаслідок цього два методи, порівняні за одним показником, часто виявляються непорівнянними за іншим, а загальний висновок про перевагу одного підходу над іншим залишається якісним.

Практичним підґрунтям для узагальнення слугує кваліфікаційна робота [2], у якій експериментально зіставлено чотири типи словників проти єдиного набору з тридцяти еталонних паролів на алгоритмах MD5, SHA-256, bcrypt та Argon2. Отримані там показники результативності та часу компрометації придатні для формального узагальнення, проте самі по собі не зводяться до спільної шкали, придатної для порівняння методів різної природи.

Метою роботи є дослідження актуальних векторів атак на паролльні системи та побудова моделі обчислювальної складності й параметра ефективності, що поєднує вартість підбору з імовірністю успіху. Наукова новизна полягає у формалізації складності окремо для класів атак з урахуванням тривалості хешування як множника вартості та у введенні параметра ефективності, який дає змогу зіставити прицільні й масові методи підбору в межах спільного апарату з урахуванням стійкості алгоритму зберігання.

МОДЕЛЬ ОБЧИСЛЮВАЛЬНОЇ СКЛАДНОСТІ ВИДІВ АТАК

Для побудови єдиної метрики порівняння необхідно спершу формалізувати саме поняття вартості атаки, спільне для методів різної природи. Будь-яку атаку підбору, незалежно від способу формування словника, можна подати як послідовну перевірку записів до першого збігу з цільовим паролем [3].

Кожна перевірка складається з обчислення хешу чергового запису та його порівняння з еталонним значенням, причому обчислення хешу є домінантною за часом операцією, тоді як порівняння виконується практично миттєво.

Повна вартість атаки визначається двома незалежними чинниками, тим, скільки записів довелося перевірити до успіху, і тим, скільки коштує обчислення одного хешу [4]. Перший чинник відображає якість словника, другий – властивості алгоритму зберігання. Формалізуючи це спостереження, запишемо обчислювальну складність атаки у вигляді добутку.

$$C = R \cdot t_h, \quad (1)$$

де C – обчислювальна складність атаки, виражена в секундах реального часу; R – позиція знайденого пароля у словнику, тобто кількість перевірених записів до влучення; t_h – тривалість одного хешування обраним алгоритмом за заданих параметрів, с.

Обидва множники виразу (1) мають чітку інтерпретацію. Величина R повністю визначається видом атаки та якістю прив'язки словника до цілі, прицільний словник, у якому правильний пароль стоїть на низькій позиції, дає мале R , тоді як неприцільний масив вимагає перебору значної частини обсягу. Величина t_h , навпаки, не залежить від словника й задається виключно алгоритмом хешування та його параметрами складності, швидкі алгоритми дають t_h порядку часток мікросекунди, навмисно повільні – порядку секунд. Такий розподіл ролей є принциповим для подальшого аналізу, оскільки він дозволяє окремо досліджувати вплив словника через R та вплив системи зберігання через t_h . Нижче вираз (1) конкретизується для кожного з п'яти класів атак шляхом визначення відповідного R , а сам характер цього визначення виявляє глибоку різницю між прицільними та неприцільними методами.

Атака повного перебору

Повний перебір є вихідним і найпростішим за логікою методом підбору, що не робить жодних припущень про структуру пароля [5]. Він послідовно генерує всі можливі комбінації символів заданого алфавіту, починаючи з найкоротших і поступово збільшуючи довжину, доки не буде знайдено збіг або вичерпано весь простір. Оскільки метод не спирається ні на статистику мови, ні на персональні дані, усі комбінації для нього априорно рівномірні, а тому в середньому доводиться перебрати половину простору, а в найгіршому випадку – увесь простір повністю [6]. Кількість необхідних перевірок безпосередньо дорівнює потужності простору паролів, унаслідок чого складність повного перебору набуває вигляду.

$$C_{bf} = N^L \cdot t_h, \quad (2)$$

де C_{bf} – обчислювальна складність повного перебору, с; N – потужність алфавіту, тобто кількість різних символів, доступних для формування пароля; L – довжина пароля у символах; t_h – тривалість одного хешування, с.



Ключовою і водночас руйнівною для цього методу властивістю виразу (2) є експоненційна залежність простору від довжини пароля. Кожен додатковий символ пароля збільшує кількість можливих комбінацій у стільки разів, скільки символів налічує алфавіт, тому навіть незначне подовження пароля робить перебір невідомим. Проілюструємо це на реалістичних значеннях. Для типового набору, що містить малі й великі літери, цифри та поширені спеціальні символи, алфавіт налічує 95 символів. За довжини пароля вісім символів простір становить 6 634 204 312 890 625 комбінацій – приблизно 6,6 квадрильйона. Якщо припустити, що паролі зберігаються за допомогою повільного алгоритму Argon2id із виміряним у цій роботі часом одного хешування 0,674 секунди, то повний перебір усього простору тривав би приблизно 141,8 мільйона років. Навіть на швидкому алгоритмі MD5, де одне хешування триває близько десятиміліонної частки секунди, той самий перебір зайняв би приблизно 21 рік.

Корпусна словникова атака

Корпусна атака є першим кроком від сліпого перебору до цілеспрямованого й будується на простому, але дієвому спостереженні, люди масово повторюють одні й ті самі паролі, тому величезна кількість реальних облікових записів захищена порівняно невеликою множиною популярних комбінацій [7]. Зловмисник, який зібрав такий перелік із оприлюднених витоків, може перевіряти його записи безпосередньо, застосовуючи їх без змін або з простими правилами трансформації на кшталт заміни літер цифрами. Записи корпусного словника не впорядковані за ймовірністю появи саме для конкретної цілі, оскільки корпус є універсальним і не прив'язаним до особи, тому ймовірність того, що правильний пароль стоїть на певній позиції, рівномірно розподілена й дорівнює $1/M$, а в разі відсутності потрібного запису доводиться перебрати весь обсяг [8]. Гранична складність такої атаки становить.

$$C_{corp} = M \cdot t_h, \quad (3)$$

де C_{corp} – обчислювальна складність корпусної атаки, с; M – повний обсяг корпусного словника у записах; t_h – тривалість одного хешування, с.

Порівняння виразів (2) і (3) показує масштаб виграшу, де корпусна атака скорочує кількість перевірок із величини порядку 10^{15} для восьмисимвольного пароля до обсягу самого словника, що в цій роботі становить мільйон – перший мільйон записів витоків RockYou 2009, відфільтрованих за довжиною від шести до двадцяти символів. Різниця в дев'ять порядків робить корпусну атаку принципово здійсненою там, де повний перебір безнадійний. Проте ця перевага має чітку межу застосовності. Оскільки корпус складається з популярних, але знеособлених паролів, він ефективний лише проти цілей, що обрали типову поширену комбінацію, і повністю безпорадний проти паролів, побудованих на унікальних персональних даних. У такому разі жоден із мільйона записів не збігається з ціллю, і вся обчислювальна вартість витрачається марно. Широке охоплення типових паролів за цілковитої неспроможності проти персоналізованих є визначальною характеристикою корпусного підходу, що безпосередньо підтверджується експериментом, у якому мільйонний корпус не дав жодного точного збігу з паролями, сформованими на основі особистих даних профілів.

Статистична атака на основі ланцюгів Маркова

Статистична атака є подальшим розвитком ідеї скорочення простору й відрізняється від корпусної тим, що не зберігає готового переліку паролів, а породжує нові комбінації, попередньо навчившись на реальному корпусі внутрішнім закономірностям сполучення символів [9]. Модель ланцюга Маркова ґрунтується на припущенні, що ймовірність появи чергового символу залежить не від усієї попередньої послідовності, а лише від кількох безпосередньо попередніх символів, що дозволяє компактно описати статистику мови паролів. Навчена модель генерує паролі за спаданням їхньої ймовірності, розташовуючи найтипівіші комбінації на початку словника, завдяки чому типові паролі перевіряються першими. Ймовірність конкретного пароля w у марковській моделі порядку k обчислюється як добуток умовних ймовірностей послідовних переходів між символами [10].

$$p(w) = P(c_1 \dots c_k) \cdot \prod_{i=k+1}^L P(c_i | c_{i-k} \dots c_{i-1}), \quad (4)$$

де $p(w)$ – ймовірність генерації пароля w у межах моделі; w – оцінюваний пароль; k – порядок марковської моделі, тобто кількість попередніх символів, що враховуються під час прогнозування наступного; c_i – i -й символ пароля; $P(c_1 \dots c_k)$ – ймовірність початкової послідовності з перших k символів; $P(c_i | c_{i-k} \dots c_{i-1})$ – умовна ймовірність появи i -го символу за умови k безпосередньо попередніх; добуток береться по всіх символах пароля, починаючи з $(k+1)$ -го.

Вираз (4) показує, що модель оцінює правдоподібність пароля через локальні статистичні залежності, а впорядкування записів за спаданням цієї ймовірності зменшує очікувану позицію влучення для статистично типових комбінацій, тому за однакового обсягу словника середня складність статистичної атаки нижча, ніж неупорядкованої корпусної. Однак принципове обмеження методу полягає в тому, що модель навчається на знеособленому корпусі й оперує лише загальними закономірностями мови, не маючи жодної інформації про конкретну особу. Через це для персоналізованих паролів, побудованих на



унікальному імені, даті чи їх поєднанні, статистично ймовірна комбінація не збігається з реальною ціллю, і влучення практично не настає. Складність невідомого прогону обчислюється так само, як і для корпусної атаки – множенням обсягу словника на час одного хешування, і в цій роботі обсяг статистичного словника становить п'ять тисяч записів. Статистична атака завдяки цьому переважає корпусну за щільністю ймовірних комбінацій на одиницю обсягу, але успадковує її фундаментальну неприцільність, оскільки залишається орієнтованою на середньостатистичного, а не на конкретного користувача.

Комбінаторна атака за OSINT-правилами

Комбінаторна атака здійснює якісний перехід від неприцільних методів до прицільних, оскільки вперше будує словник не із загальної статистики, а безпосередньо з персональних даних конкретної цілі, зібраних із відкритих джерел засобами розвідки за відкритими джерелами. Рушій такої атаки бере відомі про особу відомості – ім'я, дату народження, місто проживання, захоплення та їхні окремі фрагменти – і систематично комбінує їх за фіксованим набором правил трансформації, породжуючи всі правдоподібні варіанти пароля, які могла б обрати саме ця людина. Обсяг сформованого словника не є довільним параметром, а прямо впливає з кількості вихідних даних та застосованих правил, що формально виражається сумою по всіх профілях добутку кількості персональних елементів на кількість правил.

$$M_{osint} = \sum_{j=1}^n E_j \cdot K, \quad (5)$$

де M_{osint} – загальний обсяг комбінаторного словника у записах; E_j – кількість персональних елементів j -го профілю, тобто окремих даних, доступних для комбінування; K – кількість правил комбінування, серед яких вставлення роздільників, зміна регістру, leet-заміни символів на візуально подібні цифри та додавання типових суфіксів; n – кількість профілів, для яких будується словник; сума береться по всіх профілях від $j = 1$ до $j = n$.

Вираз (5) показує, від чого залежить обсяг словника, він зростає пропорційно і кількості зібраних про особу даних, і багатству набору правил, тому для десяти профілів у цій роботі сумарний обсяг становить 17 684 записи. Принципова відмінність від попередніх методів полягає в тому, що кожен запис такого словника за побудовою релевантний конкретній цілі, а не середньостатистичному користувачеві. Унаслідок цього правильний пароль розташований у переліку значно ближче до початку, ніж він стояв би у знеособленому корпусі, виміряна позиція першого влучення становить 16 312, а вартість влучення визначається її добутком на час одного хешування. Хоча ця позиція не є найнижчою серед усіх методів, вона на порядки менша за потужність простору повного перебору й дозволяє атаці залишатися дієвою навіть на повільних алгоритмах зберігання, оскільки кількість перевірок до успіху обмежена структурою профілю, а не розміром усього простору паролів. Саме прив'язка до цілі, а не обсяг словника, забезпечує результативність комбінаторного підходу, що й підтверджується найвищою серед усіх методів часткою скомпрометованих паролів.

Генеративна атака на основі великої мовної моделі

Генеративна атака є найдосконалішим із розглянутих прицільних методів і відрізняється від комбінаторного тим, що замість механічного перебору правил залучає велику мовну модель, здатну враховувати не лише формальні поєднання даних, а й глибший семантичний контекст особи. Там, де комбінаторний рушій сліпо застосовує фіксовані правила, мовна модель розуміє змістові зв'язки – наприклад, що людина із захопленням садівництвом схильна включати відповідні слова до пароля, – і завдяки цьому ранжує записи за їхньою правдоподібністю для конкретної цілі, розташовуючи найімовірніші паролі на самому початку короткого словника. Складність влучення для такого методу становить.

$$C_{llm} = R_{llm} \cdot t_h, R_{llm} \ll M_{corp} \quad (6)$$

де C_{llm} – обчислювальна складність генеративної атаки, с; R_{llm} – позиція влучення у згенерованому словнику; t_h – тривалість одного хешування, с; M_{corp} – обсяг корпусного словника, наведений для порівняння; знак $\ll$ означає «значно менше».

Співвідношення позиції влучення й обсягу корпусного словника у виразі (6) є ключовим і виражає головну перевагу підходу, попри те що генеративний словник украй малий, правильний пароль опиняється в ньому на надзвичайно низькій позиції. Виміряні значення підтверджують це якнайкраще – позиція влучення становить лише 232 за загального обсягу словника 1028 записів, тобто майже в тисячу разів меншого, ніж корпусний. Внаслідок такої мінімальної позиції влучення генеративна атака є найдешевшою за складністю на одне влучення серед усіх п'яти розглянутих методів, навіть попри те що загальне охоплення цілей у неї нижче, ніж у комбінаторного підходу. Відмінність між низькою позицією влучення та обмеженим охопленням має принциповий характер і не виявляється при порівнянні самих лише часток компрометації, вона стає кількісно вимірною лише через параметр ефективності, що вводиться в наступному розділі. Саме поєднання мінімальної позиції влучення з ненульовою ймовірністю успіху



робить генеративний підхід найефективнішим за витратами, відрізняючи його від комбінаторного методу, який виграє широтою охоплення, але програє позицією влучення.

Узагальнена інтерпретація складності

Зіставлення виразів (2) – (6) показує, що всі види атак зводяться до єдиної форми (1), у якій різняться лише спосіб визначення кількості перевірених записів. Для повного перебору ця кількість дорівнює всьому простору можливих комбінацій і не залежить від властивостей цілі. Для корпусної та статистичної атак вона обмежена обсягом словника, але не прив'язана до конкретної особи, тому за відсутності потрібного запису дорівнює повному обсягу. Для комбінаторної та генеративної атак це фактична позиція влучення, що визначається якістю прив'язки словника до цілі, а не його обсягом. Відмінність має принциповий характер: у неприцільних методах кількість перевірок зростає з обсягом словника, тому підвищення охоплення прямо збільшує вартість атаки без гарантії успіху, тоді як у прицільних методах позиція влучення лишається низькою навіть за незначного обсягу, оскільки правильний запис розташований близько до початку.

Наблизити правильний запис до початку вдається не нарощуванням кількості записів, а підвищенням їхньої релевантності цілі, і за незмінної тривалості одного хешування складність визначається виключно позицією влучення, тому масштаб масиву перестає бути мірою його небезпеки.

ПАРАМЕТР ЕФЕКТИВНОСТІ АТАКИ

Побудована в попередньому розділі модель складності описує вартість атаки, тобто обсяг обчислень, потрібний для досягнення результату, але сама по собі вона ще не дозволяє повноцінно порівнювати методи, оскільки не враховує їхньої результативності. Атака може бути дешевою, але майже завжди невдалою, або дорогою, але надійною, і без урахування ймовірності успіху зіставлення за самою лише складністю було б неповним. Тому для повного опису атаки необхідно поєднати її вартість із часткою цілей, які вона реально компрометує. Почнемо з граничного випадку невдалої атаки. Якщо словник не містить правильного пароля, позиція влучення не визначена, і природно оцінювати витрачені ресурси повним проходом усього словника, оскільки зловмисник перебирає всі записи до останнього, перш ніж переконатися в невдачі. Складність такого повного проходу становить.

$$C_{fail} = M \cdot t_h, \quad (7)$$

де C_{fail} – обчислювальна складність невдалої атаки за повного проходу словника, с; M – повний обсяг словника у записах; t_h – тривалість одного хешування, с.

Вираз (7) виконує важливу розрізняльну функцію, він показує, що дві однаково безуспішні атаки можуть суттєво відрізнитися за ціною провалу. Безрезультатний прогін мільйонного корпусу коштує в сотні разів дорожче, ніж прогін словника з кількох тисяч записів, хоча обидва не дають жодного зламаного пароля. Різниця чисельно виражає марність нарощування обсягу неприцільного словника, збільшення кількості записів підвищує вартість атаки, не гарантуючи успіху. Далі формалізуємо результативність. Ймовірність успіху атаки оцінюється емпірично як відносна частота компрометації, тобто відношення кількості зламаних паролів до загального числа цілей.

$$P = \frac{S}{T}, \quad (8)$$

де P – емпірична ймовірність успіху атаки; S – кількість підібраних паролів; T – загальна кількість цільових паролів.

Вираз (8) задає пряму міру охоплення: він показує, яку частку цілей метод здатний скомпрометувати, і саме це відношення традиційно наводять як основний показник результативності словникових атак. Однак сама по собі ймовірність успіху не враховує ціни досягнутого результату, а сама по собі складність не враховує його ймовірності, тому жодна з цих величин окремо не є повною характеристикою методу. Об'єднавши їх, уведемо параметр ефективності як обсяг результату, що припадає на одиницю обчислювальних витрат.

$$E = \frac{P}{C} = \frac{P}{R \cdot t_h}, \quad (9)$$

де E – параметр ефективності атаки; P – ймовірність успіху; C – обчислювальна складність, с; R – позиція влучення; t_h – тривалість одного хешування, с.

Вираз (9) поєднує обидві сторони атаки в єдиному числі й має прозорий зміст, ефективність тим вища, чим більший результат дає метод і чим менше обчислень для цього потребує. Найефективнішою є атака, що досягає високої ймовірності успіху за малих витрат, тобто така, у якій правильний пароль знаходиться на низькій позиції в прицільному словнику. Особливо важливою є структура знаменника. Оскільки тривалість одного хешування входить у нього як спільний для всіх без винятку видів атак множник, заміна швидкого алгоритму зберігання на повільний зменшує ефективність усіх методів



пропорційно – в однакову кількість разів, не змінюючи їхнього відносного порядку. Виявлена закономірність означає, що вибір алгоритму зберігання діє як єдиний масштабувальний чинник, який рівномірно знижує результативність будь-якої атаки, тоді як відмінності між самими словниками цілком зосереджені в позиції влучення. Тим самим вираз (9) формально відокремлює два незалежні аспекти захисту – вплив словника, що діє через позицію влучення, та вплив системи зберігання, що діє через вартість хешування, – і саме ця роздільність надає параметру діагностичної цінності.

Розгляд граничних випадків виразу (9) уточнює його поведінку. За ймовірності успіху, що прямує до нуля, ефективність прямує до нуля незалежно від значення складності, що відповідає повністю невдалій атаці. Якщо метод не компрометує жодної цілі, то жодні, навіть мінімальні, обчислювальні витрати не є виправданими, і ефективність дорівнює нулю.

Такий випадок безпосередньо реалізується для корпусного та статистичного словників, які в експерименті не дали жодного точного збігу. Протилежну межу дає ситуація, коли правильний пароль стоїть на першій позиції словника й вгадується з першої спроби. У цьому випадку ефективність досягає максимального можливого значення, що дорівнює ймовірності успіху, поділений на вартість одного хешування, і відповідає ідеалу прицільної атаки, у якій обчислювальні витрати зведені до абсолютного мінімуму.

Реальні прицільні методи наближаються до цієї межі тим більше, чим точніше словник ранжований під конкретну ціль. Проміжні позиції влучення забезпечують плавний перехід між нульовою та максимальною ефективністю. Окрім описової функції, параметр (9) має і безпосереднє прикладне значення для захисту. Оскільки метою оборонця є мінімізація ефективності будь-якої можливої атаки, зі структури виразу випливають рівно два незалежні важелі впливу. Перший – зменшення ймовірності успіху, що досягається усуненням прив'язки паролів до персональних даних, унаслідок чого прицільні словники втрачають матеріал для побудови й їхнє охоплення падає. Другий – одночасне збільшення позиції влучення та вартості хешування, тобто підвищення позиції влучення через непередбачуваність пароля та зростання вартості хешування через вибір навмисно повільного алгоритму. Запропонований параметр не лише впорядковує атаки за їхньою результативністю, а й прямо вказує на дві точки прикладання захисних заходів, що узгоджується з пошаровою стратегією захисту.

Числовий приклад обчислення параметра ефективності

Обчислення за виразом (9) виконано на реальних вимірюваннях зі словників OSINT і LLM на алгоритмі Argon2id [11]. Для комбінаторного словника ймовірність успіху становить 66,7 %, правильний пароль знайдено на позиції 16 312, а одне хешування триває 0,674 секунди. Складність, за виразом (1), дорівнює добутку цих двох чисел – приблизно 10 994 секунди, тобто 3,05 години. Поділивши ймовірність успіху на цю складність, отримуємо параметр ефективності приблизно 0,0000607.

Для генеративного словника ймовірність успіху нижча – 3,3 %, проте позиція влучення значно менша, 232. Складність становить приблизно 156,4 секунди, тобто 2,6 хвилини, а параметр ефективності – приблизно 0,000211. Хоча ймовірність успіху LLM-словника у двадцять разів нижча за OSINT, його ефективність вища приблизно в 3,5 раза, оскільки позиція влучення менша майже у сімдесят разів. Приклад наочно ілюструє центральну властивість параметра (9), він упорядковує методи не за охопленням цілей, а за вартістю одного результативного влучення, що й пояснює переваги генеративного підходу, приховані при зіставленні самих часток компрометації.

АРХІТЕКТУРА СИСТЕМИ ТА МЕТОДИКА ЕКСПЕРИМЕНТУ

Для перевірки моделі використано програмну систему, реалізовану за модульним принципом із чотирьох функціональних шарів та оркестратора, що керує їхнім послідовним виконанням. Шар генерації завантажує профілі та будує словники, шар хешування обчислює еталонні хеші всіма алгоритмами з унікальною сіллю, шар атаки проганяє словники проти хешів, а аналітичний шар перетворює результат із бінарного «зламано чи ні» на реальний час до зламу. Кожен обчислювальний модуль отримує дані через параметри й повертає результат у вигляді типізованого об'єкта, без читання файлів, глобального стану чи побічних ефектів усередині логіки, а запис на диск відбувається лише наприкінці. Принцип чистих функцій і незмінних структур гарантує, що всі показники звіту базуються на даних одного запуску, без ризику змішування результатів різних прогонів, що є необхідною умовою відтворюваності експерименту.

Параметри хешування задано так, щоб забезпечити коректне зіставлення алгоритмів у межах реалістичного часового бюджету експерименту. Для bcrypt застосовано коефіцієнт вартості $\text{cost} = 12$ (4096 ітерацій), що відповідає типовому промислового налаштуванню. Для Argon2id використано варіант із вимогою до пам'яті $\text{memory_cost} = 64$ МБ, часовим параметром $\text{time_cost} = 1$ та ступенем паралелізму $\text{parallelism} = 2$. Часовий параметр свідомо занижено відносно рівня, придатного для промислового навантаження, щоб укластися в час обчислювального експерименту, отримані внаслідок цього значення



часу до зламу є нижньою межею реального захисту, а за рекомендованих параметрів вони зросли б пропорційно, не змінюючи відносного порядку словників.

Вхідними даними слугують десять профілів, кожен із яких містить ім'я, рік народження, місто та захоплення. До кожного профілю прив'язано три паролі різної складності, що разом дає $T = 30$ цільових паролів. Еталонна база сформована так, щоб охопити різні рівні передбачуваності, прості паролі містять один персональний елемент, наприклад ім'я з роком народження, середні поєднують кілька елементів із роздільником або зміною регістру, складні мають менш очевидну структуру з маскуванням чи нестандартним поєднанням. Такий розподіл дозволяє оцінити, як кожен вид словника справляється з цілями зростаючої складності, а не лише з типовими паролями. Профілі побудовані за реальними закономірностями формування паролів, задокументованими в аналізі мільярдів записів [3], тому відтворюють типову поведінку користувачів, не будучи прив'язаними до конкретних осіб. Кожен пароль хешується всіма чотирма алгоритмами незалежно, що дає шістнадцять комбінацій «словник – алгоритм» і забезпечує повне покриття простору порівняння.

Кожен із чотирьох словників представляє окремий підхід до генерації записів. Корпусний містить перший мільйон записів реального витоку без перетворень.

Статистичний згенеровано моделлю ланцюгів Маркова, навченою на першій половині корпусу, що породжує нові комбінації, відсутні в корпусі, поділ даних на навчальну й тестову частини запобігає перенаванчання моделі та усуває перевірку словника на власних даних.

Комбінаторний побудовано рушієм OSINT-правил, який систематично поєднує персональні дані профілю без застосування нейромереж. Генеративний синтезовано локальною великою мовною моделлю, якій профіль передається через структурований запит без донавчання.

Результат атаки обробляється на трьох рівнях глибини, точний збіг рядка, коли запис буквально збігається з паролем, контекстна релевантність, коли запис містить пароль як підрядок без урахування регістру, та гібридна атака, коли до близького запису застосовуються типові правила мутації реальних інструментів зламу. Для кожного знайденого пароля обчислюється його позиція у словнику, помножена на час перевірки одного запису, що дає реальний час до зламу.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Результати тривірневого аналізу компрометації наведено в таблиці 1.

Таблиця 1

Результати компрометації тридцяти еталонних паролів

Словник	Обсяг	Точний збіг	Релевантність	Гібридна
OSINT-правила	17 684	66,7 %	66,7 %	66,7 %
Генеративна (LLM)	1 028	3,3 %	13,3 %	13,3 %
RockYou	1 000 000	0 %	30 %	0 %
Markov	5 000	0 %	0 %	0 %

Розподіл результатів між словниками виявляє закономірність, що суперечить інтуїтивному уявленню про перевагу обсягу. Мільйонний корпус RockYou, найбільший серед випробуваних, дав нульовий точний збіг, тоді як його релевантність у 30 % свідчить про часткові перетини на кшталт спільних імен без цифрового доповнення. Такий результат означає, що корпус містить лінгвістично близькі, але не тотожні цілям записи, які не доводяться до збігу простими правилами, оскільки бракує саме персонального цифрового компонента. Статистичний словник Маркова не дав ані точних збігів, ані релевантності, що пояснюється його знеособленою природою, модель відтворює частоти символічних переходів загального корпусу, але не має інформації про конкретну особу. Комбінаторний та генеративний словники, навпаки, прив'язані до профілю, тому навіть за малого обсягу влучають у персоналізовані цілі. Комбінаторний OSINT-словник, у 57 разів менший за корпус, скомпрометував дві третини паролів, причому всі близькі влучення доведено до точних елементарних правилами регістру, а генеративний словник поступився за форматною точністю, але показав вищу контекстну обізнаність, підвищивши релевантність до 13,3 %. Приклади задокументованих зломів наведено в таблиці 2.

Таблиця 2

Приклади задокументованих гібридних зломів

Словник	Запис-джерело	Правило мутації	Результат
OSINT	SVITLANA81	нижній регістр	svitlana81
OSINT	SVITLANA_0807	з великої літери	Svitlana_0807
LLM	Vlad1997Gardening	підрядок + регістр	vlad1997



Часові характеристики алгоритмів хешування для перевірки мільйонної вибірки наведено в таблиці 3.

Таблиця 3

Час перевірки мільйонної вибірки та тривалість одного хешування

Алгоритм	Час на 10^6 записів	t_h на запис, с
MD5	0,66 с	$6,6 \cdot 10^{-7}$
SHA-256	0,65 с	$6,5 \cdot 10^{-7}$
bcrypt	58,5 діб	5,05
Argon2id	7,8 діб	0,674

Швидкі алгоритми обробляють мільйон записів менш ніж за секунду, тоді як навмисно повільні розтягують той самий перебір на тижні. Параметри складності Argon2id задано нижче рекомендованого рівня, тому наведені значення відповідають нижній оцінці стійкості. Підставлення t_h у вирази (2) – (6) дає складність кожного виду атаки, зведену в таблицю 4.

Таблиця 4

Обчислювальна складність видів атак на алгоритмі Argon2id

Вид атаки	Формула	R	Складність C
Повний перебір	$N^L \cdot t_h$	$6,6 \cdot 10^{15}$	теоретична межа
Корпусна (RockYou)	$M \cdot t_h$	10^6	7,80 діб
Статистична (Markov)	$M \cdot t_h$	$5 \cdot 10^3$	56,2 хв
Комбінаторна (OSINT)	$R \cdot t_h$	16 312	3,05 год
Генеративна (LLM)	$R \cdot t_h$	232	2,6 хв

Повний перебір наведено як теоретичну верхню межу й не залучено до порівняння ефективності, для невдалих атак наведено складність повного проходу за виразом (7). Таблиця 4 демонструє діапазон складності близько п'ятнадцяти порядків – від практично нездійсненого повного перебору до хвилин генеративної атаки на тому самому алгоритмі.

На однаковому рубежі зберігання вирішальною є позиція влучення, тобто прицільність словника, а не обсяг обчислень взагалі. Значення параметра ефективності для чотирьох вимірних словників на всіх алгоритмах хешування наведено в таблиці 5.

Таблиця 5

Значення параметра ефективності E словників за алгоритмами хешування

Словник	P	MD5	SHA-256	bcrypt	Argon2
OSINT	0,667	$6,20 \cdot 10^1$	$6,29 \cdot 10^1$	$8,09 \cdot 10^{-6}$	$6,07 \cdot 10^{-5}$
LLM	0,033	$2,16 \cdot 10^2$	$2,19 \cdot 10^2$	$2,81 \cdot 10^{-5}$	$2,11 \cdot 10^{-4}$
RockYou	0	0	0	0	0
Markov	0	0	0	0	0

Аналіз таблиці 5 виявляє три закономірності. Ненульову ефективність мають лише прицільні словники, тоді як корпусний і статистичний дають нульову ефективність на всіх алгоритмах незалежно від обсягу, а вираз (7) зберігає між ними різницю за вартістю: безрезультатний прогін мільйонного RockYou обходиться приблизно у двісті разів дорожче за прогін п'ятитисячного Markov. Генеративний словник має вищу ефективність за комбінаторний на всіх алгоритмах, попри нижчу ймовірність успіху: на Argon2id його ефективність становить $2,11 \cdot 10^{-4}$ проти $6,07 \cdot 10^{-5}$, оскільки правильний пароль знайдено на позиції 232 проти 16 312. Комбінаторний метод вигідніший за охопленням цілей, а генеративний – за витратами на окреме влучення, і зіставлення самих часток компрометації цю відмінність приховує, тоді як параметр ефективності робить її кількісною. Перехід від швидких алгоритмів до повільних знижує ефективність кожної атаки приблизно на шість порядків, проте відносний порядок словників зберігається, бо тривалість одного хешування є спільним множником. Наведені в таблиці значення відповідають покроковому обчисленню, поданому в підрозділі 2.1.

Аналіз впливу окремих алгоритмів хешування

Швидкі алгоритми MD5 та SHA-256 дають майже однакову вартість перевірки, оскільки обидва спроектовані для миттєвого обчислення без навмисного уповільнення, різниця в соті частки залежить від реалізації, а не від криптографічної стійкості, і для задачі зберігання паролів несуттєва. Обидва уможливають перевірку мільйонного словника за частку секунди, що робить офлайн-перебір тривіальним і будь-який прицільний словник – надзвичайно ефективним, тому швидкі алгоритми непридатні для зберігання паролів незалежно від їхньої довжини. Повільні алгоритми bcrypt та Argon2id



підвищують вартість перевірки на шість-сім порядків. За вимірних параметрів bcrypt дає більшу тривалість одного хешування (5,05 с) за Argon2id (0,674 с), унаслідок чого повний перебір мільйонного словника на bcrypt триває 58,5 діб проти 7,8 діб на Argon2id. Розбіжність є наслідком конкретних параметрів складності, свідомо знижених для Argon2id, а не більшої швидкості алгоритму, справжня перевага Argon2id полягає у вимогах до оперативної пам'яті, які знецінюють паралельні обчислення на графічних процесорах і спеціалізованих інтегральних схемах. Тоді як bcrypt навантажує переважно процесор і піддається апаратному прискоренню, пам'ятевимогливі Argon2id зберігає стійкість проти масштабованих апаратних атак, тому за однакового цільового часу забезпечує вищий реальний захист. Підсумкове зіставлення реального часу до зламу на найстійкішому алгоритмі Argon2id наведено в таблиці 6.

Таблиця 6

Реальний час до зламу на алгоритмі Argon2id

Словник	Ціль	Ранг	Час до зламу
Генеративна (LLM)	Lyosha_0710	232	2,6 хв
OSINT-правила	svitlana81	16 312	3,0 год
RockYou	немає збігу	–	7,8 діб

Наведені значення підтверджують головний висновок, розмір словника не визначає успішності атаки. Генеративний словник із 1 028 записів, майже в тисячу разів менший за мільйонну вибірку, знаходить пароль на 232-й позиції за лічені хвилини, тоді як сам мільйонний масив за повного перебору не дає жодного результату за тиждень безперервної роботи. Для оцінки впливу вибору повільного алгоритму на всі види атак у таблиці 7 зіставлено складність кожного словника на двох повільних алгоритмах.

Таблиця 7

Складність видів атак на повільних алгоритмах bcrypt та Argon2id

Словник	bcrypt ($t_h = 5,05$ с)	Argon2id ($t_h = 0,674$ с)
Генеративна (LLM)	19,5 хв	2,6 хв
Статистична (Markov)	7,0 год	56,2 хв
Комбінаторна (OSINT)	22,9 год	3,05 год
Корпусна (RockYou)	58,5 діб	7,80 діб

Дані таблиці 7 показують, що за вимірних параметрів bcrypt дає більшу тривалість перебору за Argon2id для кожного словника, проте це не свідчить про більшу захисну спроможність bcrypt. Різниця зумовлена конкретними параметрами складності, а не архітектурою алгоритму, і за належного налаштування Argon2id досягає будь-якого цільового часу. Вирішальною перевагою Argon2id лишається його пам'ятевимогливість, яка не відображається у самому значенні t_h , але знецінює апаратне прискорення атаки, чого послідовний за пам'яттю bcrypt не забезпечує. Модель складності, відповідно, коректно ранжує словники, тоді як вибір між повільними алгоритмами потребує врахування додаткового чинника стійкості до спеціалізованого обладнання.

ЗАСТОСУВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ОПТИМІЗАЦІЇ ПІДБОРУ

Розвиток методів підбору відбувався від статистичних моделей до нейромережових. На початковому етапі використовувалися ймовірнісні ланцюги Маркова, що генерують комбінації за частотою символічних послідовностей. Упорядкований марковський енумератор розташовує паролі з найвищою ймовірністю на початку словника для прискорення підбору [10]. Гібридні архітектури доповнюють статистичний генератор нейромережовим дискримінатором, який після генерації кожної комбінації обчислює оцінку її правдоподібності й відсіює малоімовірні записи ще до початку атаки. Оцінка правдоподібності згенерованого пароля визначається функцією дискримінатора [10].

$$s = D(x), \quad (10)$$

де s – оцінка ймовірності належності зразка до розподілу реальних паролів; D – навчена функція дискримінатора; x – згенерований пароль, поданий на вхід дискримінатора.

Записи, оцінка яких перевищує встановлений експериментально поріг, потрапляють до фінального словника, а решта відкидається, унаслідок чого середня якість записів зростає, а частка марних перевірок під час атаки зменшується. Такий відбір дозволяє скоротити обсяг словника без втрати результативності, що прямо впливає на позицію влучення й, відповідно, на складність атаки в подальшій моделі.



Прагнучи повністю усунути потребу в ручних правилах трансформації, дослідники застосували генеративно-змагальні мережі [12], які самостійно вивчають розподіл реальних паролів безпосередньо з масивів витоків. Модель складається з двох протилежних нейромереж – генератора, що синтезує нові паролі з випадкового шуму, та дискримінатора, що намагається відрізнити їх від справжніх. Навчання формалізується як мінімаксна задача, у якій генератор мінімізує, а дискримінатор максимізує спільну цільову функцію [12].

$$\min_G \max_D V(D, G) = E[\log D(x)] + E[\log(1 - D(G(z)))], \quad (11)$$

де $\min_G$ та $\max_D$ – операції мінімізації за параметрами генератора й максимізації за параметрами дискримінатора; $V(D, G)$ – цільова функція змагальної гри між генератором і дискримінатором; D – дискримінатор; G – генератор, що створює пароль $G(z)$ із вектора шуму; E – математичне сподівання по відповідному розподілу; $D(x)$ – оцінка дискримінатором справжності реального зразка; x – реальний пароль із навчальної вибірки; $G(z)$ – пароль, синтезований генератором; z – вектор випадкового шуму на вході генератора.

У процесі змагання генератор поступово навчається відтворювати статистичні та структурні закономірності людського словотворення, недоступні для експертних правил, і здатний породжувати практично необмежену кількість нових комбінацій. Завдяки цьому генеративний підхід стає джерелом записів, яких немає в жодному наявному витокі, що розширює охоплення простору паролів за межі готових корпусів.

Наступний етап пов'язаний із великими мовними моделями на основі архітектури Transformer, які, на відміну від суто статистичних методів, обробляють природну мову й уловлюють семантичні зв'язки. Такі моделі обчислюють ймовірність пароля як добуток умовних ймовірностей появи кожного символу за умови всіх попередніх, що відповідає авторегресійному породженню послідовності [13].

$$P(w) = \prod_{i=1}^L P(c_i | c_1 \dots c_{i-1}), \quad (12)$$

де $P(w)$ – загальна ймовірність згенерованого пароля; w – пароль; c_i – його i -й символ; $P(c_i | c_1 \dots c_{i-1})$ – умовна ймовірність появи i -го символу за умови всіх попередніх; L – довжина пароля у символах, добуток береться по всіх символах від $i = 1$ до $i = L$.

Керована генерація дозволяє задавати обмеження щодо довжини чи структури, а для цільового підбору породження стає умовним не лише відносно попередніх символів, а й відносно персональних даних жертви та історії її старих паролів [13]. Завдяки цьому мовна модель синтезує вузький словник, у якому найімовірніші для конкретної особи паролі розташовані на початку, що безпосередньо зменшує позицію влучення й вартість атаки.

Ефективність описаних підходів підтверджується експериментально. Порівняння впорядкованого марковського еnumerатора, генеративно-змагальної мережі та гібридної архітектури під час генерації словників зростаючого обсягу наведено в таблиці 8.

Таблиця 8

Порівняння результативності моделей генерації паролів на базі RockYou

Обсяг генерації	Марковський, %	Змагальний, %	Гібридний, %
10^5	0,51	0,06	0,57
10^6	2,69	0,53	3,04
10^7	10,09	3,68	12,63
10^8	26,62	13,40	28,29
10^9	49,88	28,10	51,91

Дані таблиці 8 свідчать, що гібридна архітектура перевершує кожен із базових підходів на всіх обсягах генерації, на десяти мільйонах записів вона покращує частку успішних збігів на 25,18 % відносно чистого марковського методу [10]. Результат підтверджує, що поєднання ймовірнісного генератора з нейромережним відбором дає вищу результативність, ніж будь-який компонент окремо. Проте всі ці методи оцінені на масових знеособлених вибірках, тому їхня ефективність проти персоналізованих цілей з обмеженою кількістю спроб залишається окремою величиною, яку й досліджує ця робота через параметр ефективності.

Зіставлення таблиць 5 і 8 проливає світло на межі суто генеративного вдосконалення словника. Приріст на 25,18 % у таблиці 8 отримано за рахунок кращого відбору серед знеособлених записів, тобто без жодної прив'язки до конкретної особи, і вимірюється в тих самих одиницях частки збігів, що й корпусний та статистичний словники цієї роботи – обидва з нульовим результатом проти персоналізованих цілей. Перехід від знеособленої генерації до цільового підбору за OSINT-правилами чи великою мовною моделлю дає натомість якісно інший порядок виграшу, не десятки відсотків у межах одного механізму



генерації, а перехід від нульової до 66,7-відсоткової результативності. Різниця підтверджує, що для персоналізованих цілей з обмеженим бюджетом спроб прив'язка словника до профілю жертви важить більше, ніж будь-яке вдосконалення алгоритму генерації в межах знеособленого підходу.

Порівняльні дані для десяти тематичних масивів, що зіставляють базовий словник Ignis-10M із доповненим генеративною моделлю PassGPT словником, показують стабільний, хоча і скромний приріст, частка записів, знайдених виключно завдяки генеративній моделі, коливається від 0,18 % до 0,88 % залежно від тематичної категорії, у середньому близько 0,48 % [13]. Незначний відсотковий приріст не применшує його значення, кожен запис є паролем, недосяжним для традиційного словника через нетипову структуру чи вузькоспецифічний контекст, тому в масштабах реальних витоків навіть частка менше одного відсотка перетворюється на тисячі додатково скомпрометованих облікових записів.

Цільовий підбір із залученням персональних даних

Сліпий перебір виявляється неефективним проти конкретних цілей за жорсткого обмеження кількості спроб, тому оптимізація компрометації досягається алгоритмами цільового підбору, які використовують схильність користувачів включати в паролі персональну ідентифікаційну інформацію. Завдання такого підбору полягає у максимізації ймовірності успішного зламу за обмежений бюджет спроб, що формалізується цільовою функцією [14].

$$P_{succ} = \sum_{i=1}^q P(pw_i | PII), \quad (13)$$

де P_{succ} – сумарна ймовірність успішної компрометації облікового запису; q – максимально допустима кількість спроб, тобто бюджет атаки; pw_i – i -й пароль-запис словника, відсортований за спаданням ймовірності; PII – відомі персональні дані цілі; $P(pw_i | PII)$ – умовна ймовірність того, що правильним є i -й запис за умови відомих персональних даних; сума береться по перших q записах.

Вираз (13) показує, що результативність цільової атаки визначається не обсягом словника, а якістю ранжування, чим точніше найімовірніші для конкретної особи паролі зміщені до початку переліку, тим вища сумарна ймовірність за фіксованого бюджету q .

Моделі на базі персональних даних генерують унікальні комбінації з вищою ефективністю, а інтеграція історії старих паролів засобами глибокого навчання досягає високої результативності за обмеженої кількості спроб [5]. Саме цю логіку реалізують комбінаторний та генеративний словники, розглянуті вище в експериментальній частині роботи.

ТЕОРЕТИЧНІ ЗАСАДИ ОЦІНЮВАННЯ СТІЙКОСТІ

Здійснене вище вимірювання реальної вартості атак спирається на позицію влучення та вартість хешування, що суттєво відрізняється від класичного способу оцінювання стійкості пароля, який історично передував емпіричним методам і досі широко застосовується на практиці. Традиційно надійність пароля виражають через інформаційну ентропію – величину, що визначає теоретичну кількість рівноймовірних варіантів, серед яких прихований секрет, і тим самим характеризує невизначеність, яку має подолати злоумисник. Чим більше різних паролів однаково правдоподібні, тим складніше вгадати правильний, і саме цю ідею формалізує міра ентропії. Для пароля довжини L , сформованого над алфавітом потужності N , за умови рівноймовірного вибору кожного символу базова ентропія обчислюється за формулою Шеннона [3].

$$H = \log_2(N^L), \quad (14)$$

де H – ентропія пароля, виражена в бітах; N – потужність алфавіту, тобто кількість різних символів, доступних для формування пароля; L – довжина пароля у символах; N^L – загальна кількість можливих комбінацій, що утворюють простір паролів.

Логарифмічна природа виразу (14) має наочну інтерпретацію, кожен додатковий біт ентропії подвоює кількість рівноймовірних варіантів, тому пароль із ентропією 40 бітів відповідає приблизно 10^{12} комбінаціям, а з 60 бітами – вже понад 10^{18} . Властивість робить ентропію зручною єдиною шкалою для порівняння паролів різної довжини й складу алфавіту, чим і зумовлена її популярність у парольних політиках. Проте формула (14) містить критичне припущення, що обмежує її застосовність, вона передбачає, що кожен символ обирається випадково, незалежно від інших і з рівномірного розподілу. Реальні користувачі так не чинять. Вони будують паролі з осмислених фрагментів – власних імен, дат народження, словникових слів, клавіатурних послідовностей, – унаслідок чого справжній розподіл паролів украй нерівномірний, невелика частка комбінацій зустрічається надзвичайно часто, тоді як переважна більшість теоретично можливих паролів не використовується ніколи. Через це фактична невизначеність людського пароля значно нижча за обчислену за формулою (14), і базова ентропія систематично й суттєво завищує реальну стійкість.



Усвідомлення цієї вади привело до появи уточнених метрик, що враховують нерівномірність реального розподілу. Однією з таких є очікувана ентропія, яка набуває значень у діапазоні від нуля до одиниці й показує частку простору пошуку, що її зломиснику доведеться в середньому перебрати для успішного вгадування, зважуючи кожен пароль на його фактичну ймовірність появи [4]. На відміну від базової ентропії, ця міра відображає, що атака на практиці зосереджується на найімовірніших комбінаціях і рідко доходить до малоімовірних, і завдяки цьому дає ближчу до реальності оцінку. Проте навіть уточнені ентропійні метрики залишаються апіорними, вони описують властивості розподілу паролів самі по собі, але не залежать від того, яким алгоритмом пароль зберігається, і не виражають фактичної вартості перебору в одиницях часу. Для зломисника ж вирішальним є саме реальний час до зламу, який визначається спільно і структурою словника, і швидкістю алгоритму хешування. Саме цей розрив між абстрактною ентропійною оцінкою та конкретною часовою вартістю підбору обґрунтовує перехід від апіорних метрик стійкості до безпосереднього вимірювання складності через позицію влучення. Саме тому в моделі, представленій у розділі 1, основною величиною слугує не теоретична ентропія H , а виміряна позиція влучення R , яка, на відміну від ентропії, безпосередньо поєднується з вартістю хешування в єдину вимірювану характеристику складності атаки.

Отримані результати визначають пріоритет заходів захисту за їхнім виміряним впливом на стійкість. Найбільший приріст дає зберігання паролів навмисно повільними ітеративними алгоритмами, передусім Argon2id, з унікальною криптографічною сіллю для кожного запису. Унікальна сіль унеможливорює застосування заздалегідь обчислених таблиць і змушує атакувати кожен запис окремо, а перевага Argon2id над старішим bcrypt полягає у вимогах до оперативної пам'яті, що знецінює перевагу спеціалізованого обладнання, атака на графічних процесорах, яка багаторазово пришвидшує перебір швидких алгоритмів, втрачає ефективність проти алгоритму, кожна спроба якого потребує значного обсягу пам'яті.

Другим рівнем є політика формування паролів, що зміщує акцент із формальної складності на довжину та відмову від паролів, похідних від публічної особистої інформації. Контекстні словники досягають високої компрометації саме тому, що користувачі складають паролі з імен, дат народження та інтересів, а випадкові паролі, згенеровані паролйними менеджерами, не містять семантичних фрагментів, на яких будуються цільові словники, і повертають атаку до неприцільного перебору, дорогого навіть на швидких алгоритмах.

Третім рівнем є обмеження швидкості спроб входу, що робить короткий цільовий словник непридатним для підбору в реальному часі, перевірка нового пароля за списками раніше скомпрометованих облікових даних, і регулярне звіряння збережених паролів із новими витоками.

Четвертим рівнем є багатофакторна автентифікація, яка додає незалежний від пароля захист і знижує наслідки компрометації, оскільки навіть розкритий пароль не відкриває доступу без другого фактора, причому перевагу надають стійким до фішингу факторам. Жоден окремий механізм не забезпечує повної безпеки, тому надійна оборона будується пошарово, а захід, що найбільше підвищує вартість атаки, розглядається першим.

ОБМЕЖЕННЯ МОДЕЛІ

Запропонована модель має низку обмежень, що окреслюють межі її застосовності. Ймовірність успіху P оцінена на вибірці з тридцяти цілей, тому її абсолютні значення мають ілюстративний характер, водночас різниця в результатах на порядки робить якісний висновок про перевагу прицільності стійким до розміру вибірки. Позиція влучення R для повільних алгоритмів отримана екстраполяцією, тривалість одного хешування виміряна безпосередньо як стабільна константа алгоритму й помножена на позицію влучення, визначену фактично, тому екстрапольованою є лише оцінка повного проходу, а не сам ранг. Параметри складності Argon2id задано нижче рекомендованого рівня для скорочення часу експерименту, унаслідок чого наведені значення часу відповідають нижній межі реального захисту, за рекомендованих параметрів складності зросла б пропорційно, посилюючи, а не послаблюючи висновок про пріоритет повільного алгоритму. Модель у поточному вигляді не враховує розподілу рангів у межах словника, обмежуючись позицією першого влучення, що є достатнім для порівняння методів, але не описує повної кривої вгадування. Порівняно з наявними ймовірнісними мірами вгадування запропонований параметр вирізняється явним урахуванням вартості одного хешування, що робить його чутливим до алгоритму зберігання, тоді як класичні ентропійні метрики оцінюють лише розподіл паролів незалежно від способу їх зберігання [15]. Подальший розвиток моделі доцільно спрямувати на врахування повного розподілу рангів, розширення вибірки профілів для статистичної значущості абсолютних оцінок та застосування до додаткових класів алгоритмів хешування з варіюванням параметрів складності.



Ще одним обмеженням є вимірювання t_h на єдиній апаратній конфігурації без урахування спеціалізованого обладнання для паралельного перебору. Для швидких алгоритмів, де одна сучасна графічна карта здатна обчислювати до 50 мільярдів хешів MD5 за секунду, реальний час до зламу на розподіленій системі буде на порядки нижчим за наведений у роботі. Для пам'ятевимогливого Argon2id таке масштабування обмежене фізичним обсягом оперативної пам'яті на пристрій, тому висновок про його перевагу над bcrypt лишається стійким і за апаратного масштабування, тоді як абсолютні значення часу для MD5 та SHA-256 варто трактувати як оцінку на одному обчислювальному вузлі, а не як межу можливостей зловмисника.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Досліджено актуальні вектори атак на пароліні системи та встановлено, що суцільний перебір поступається місцем прицільному вгадуванню за відкритими даними цілі, причому результативність визначається прицільністю словника, а не його обсягом, а найвищу ефективність за витратами на одне влучення демонструє підбір із залученням генеративних моделей. Побудовано модель обчислювальної складності словникових атак, у якій вартість підбору для п'яти видів – повного перебору (2), корпусної (3), статистичної (4), комбінаторної (5) та генеративної (6) – виражено через кількість перевірених записів і тривалість хешування, що зводить різноманітні методи до спільної величини складності (1). Введено параметр ефективності (9) як відношення ймовірності успіху до обчислювальної вартості, доповнений виразом вартості невдалого прогону (7). Експеримент підтвердив, що прицільність важить більше за обсяг, словник OSINT, у 57 разів менший за корпусну вибірку, розкрив 66,7 % паролів, тоді як мільйонний RockYou не знайшов жодного збігу. Параметр ефективності відокремлює ефективність за охопленням цілей від ефективності за витратами на влучення, приховану при зіставленні часток компрометації, та кількісно підтверджує вирішальний вплив алгоритму зберігання через спільний множник тривалості хешування. На основі результатів укладено систему заходів захисту, упорядковану за виміряним впливом, провідне місце в якій посідає зберігання паролів повільними ітеративними алгоритмами. Подальші дослідження доцільно спрямувати на перевірку моделі на ширшій вибірці профілів та на додаткових алгоритмах хешування з варіюванням параметрів складності.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Radivilova, T., Kirichenko, L., & Alghawli, A. S. (2019). Entropy analysis method for attacks detection. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 443–446). IEEE. <https://doi.org/10.1109/PICST47496.2019.9061451>
2. Заїка, П. В. (2026). Дослідження методів забезпечення пароліної безпеки на основі аналізу актуальних векторів атак [Кваліфікаційна робота бакалавра, Харківський національний університет радіоелектроніки]. Електронний архів ХНУРЕ (OpenHUB).
3. Kanta, A., Coray, S., Coisel, I., & Scanlon, M. (2021). How viable is password cracking in digital forensic investigation? Analyzing the guessability of over 3.9 billion real-world accounts. *Forensic Science International: Digital Investigation*, 37, 301186. <https://doi.org/10.1016/j.fsidi.2021.301186>
4. Rodrigues, G. A. P., Fernandes, P. A. G., Serrano, A. L. M., Ribeiro, P. H. C., & de Oliveira, A. L. I. (2025). From RockYou to RockYou2024: Analyzing password patterns across generations, their use in industrial systems and vulnerability to password guessing attacks. *Journal of Internet Services and Applications*, 16(1), 69–86. <https://doi.org/10.5753/jisa.2025.4384>
5. Nisenoff, A., Golla, M., Wei, M., Hainline, J., Szymanek, H., Braun, A., Hildebrandt, A., Christensen, B., Langenberg, D., and Ur, B. (2023). A two-decade retrospective analysis of a university's vulnerability to attacks exploiting reused passwords. In *Proceedings of the 32nd USENIX Conference on Security Symposium (SEC '23)*. USENIX Association, USA, 287, pp. 5127–5144.
6. Ou, W., Sun, C., Pang, M., Wu, Z., & Chen, J. (2026). Improving targeted password guessing attacks by using personally identifiable information and old password. *Cybersecurity* 9, 9. <https://doi.org/10.1186/s42400-025-00430-0>
7. Wang, D., Shan, X., Dong, Q., Cheng, H., & Wang, P. (2023). No single silver bullet: Measuring the accuracy of password strength meters. In *32nd USENIX Security Symposium (USENIX Security 23)* (pp. 947–964). USENIX Association.
8. National Institute of Standards and Technology. (2024). Digital identity guidelines: Authentication and lifecycle management (NIST Special Publication 800-63B-4). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.SP.800-63b-4>



9. Radivilova, T., Kirichenko, L., Tawalbeh, M., & Ilkov, A. (2021). Виявлення аномалій в телекомунікаційному трафіку статистичними методами. Електронне фахове наукове видання «Кибербезпека: освіта, наука, техніка», 3(11), 183–194. <https://doi.org/10.28925/2663-4023.2021.11.183194>
10. Jiang, J., Zhou, A., Liu, L., & Zhang, L. (2022). OMECDN: A password-generation model based on an ordered Markov enumerator and critic discriminant network. *Applied Sciences*, 12(23), 12379. <https://doi.org/10.3390/app122312379>
11. Hitaj, B., Gasti, P., Ateniese, G., Perez-Cruz, F. (2019). PassGAN: A Deep Learning Approach for Password Guessing. In: Deng, R., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds) *Applied Cryptography and Network Security. ACNS 2019. Lecture Notes in Computer Science()*, vol 11464. Springer, Cham. https://doi.org/10.1007/978-3-030-21568-2_11
12. Rando, J., Perez-Cruz, F., Hitaj, B. (2024). PassGPT: Password Modeling and (Guided) Generation with Large Language Models. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds) *Computer Security – ESORICS 2023. ESORICS 2023. Lecture Notes in Computer Science*, vol 14347. Springer, Cham. https://doi.org/10.1007/978-3-031-51482-1_9
13. Imhof, G., Kanta, A., & Scanlon, M. (2025). Context-based password cracking dictionary expansion using generative pre-trained transformers. In *2024 Cyber Research Conference - Ireland, Cyber-RCI 2024 Institute of Electrical and Electronics Engineers Inc.*. <https://doi.org/10.1109/Cyber-RCI60769.2024.10939663>
14. Wang, D., Zhang, Z., Wang, P., Yan, J., & Huang, X. (2016). Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)* (pp. 1242–1254). Association for Computing Machinery. <https://doi.org/10.1145/2976749.2978339>
15. Radivilova, T., Kirichenko, L., Panteliev, V., Mazepa, A., & Bilodid, V. (2024). Analysis of authentication methods for full-stack applications and implementation of a web application with an integrated authentication system. *Innovative technologies and scientific solutions for industries*, 3 (29), 76–90. <https://doi.org/10.30837/2522-9818.2024.3.076>

**Arkadii Snihurov**

PhD, associated professor,
Dean of the Faculty of Cybersecurity,
associate professor of the V.V. Popovskyy department of infocommunication engineering,
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
0000-0003-1355-7978
arkadii.snihurov@nure.ua

Pavlo Zaika

Master student of the Faculty of Cybersecurity,
V.V. Popovskyy department of infocommunication engineering,
Kharkiv National University of Radio Electronics, Kharkiv, Ukraine
0009-0000-4181-7429
pavlo.zaika@nure.ua

A MODEL OF COMPUTATIONAL COMPLEXITY AND EFFECTIVENESS OF PASSWORD-GUESSING ATTACKS USING OSINT AND LARGE LANGUAGE MODELS

Abstract. This paper examines modern attack vectors against password systems in a context where attempt limits have rendered brute-force attacks ineffective, while open generative models and open-source intelligence (OSINT) tools have made targeted, user-specific password guessing more cost-effective. It is shown that existing performance metrics (the proportion of compromised accounts, time to first hit, and theoretical polynomial complexity) cannot be reduced to a common scale. Consequently, methods of different natures cannot be quantitatively compared. A computational complexity model in which the cost of an attack is represented as the product of the hit position in the dictionary and the duration of a single hashing operation was proposed. The model is specified for five classes of attacks: brute-force, corpus-based, statistical (Markov chain-based), combinatorial (OSINT-based), and generative (large language model-based). An efficiency parameter is introduced as the ratio of the empirical probability of success to the computational complexity, supplemented by an estimate of the cost of a failed dictionary run. The parameter's structure separates the influence of the dictionary, which acts through the match position, from that of the storage algorithm, which acts as a common scaling factor. Experimental validation was performed on benchmark profile passwords using four dictionaries and the MD5, SHA-256, bcrypt, and Argon2id algorithms. A combinatorial OSINT dictionary, 57 times smaller than the RockYou million-sample dataset, compromised 66.7% of the passwords. The corpus and Markov dictionaries did not yield a single exact match. The generative dictionary had a lower success probability (3.3%) but found the password at the 232nd position. Its effectiveness was approximately 3.5 times that of the combinatorial dictionary, and this advantage is not apparent when comparing the compromise rates themselves. Switching to slower algorithms reduces the effectiveness of all attacks by approximately six orders of magnitude without changing their relative order. Based on the results, a layered system of protective measures is proposed, ranked by their measured impact on security.

Keywords: password authentication; password guessing attacks; dictionary attack; computational complexity; hit position; open-source intelligence (OSINT); large language models; Markov chains; hashing.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Radivilova, T., Kirichenko, L., & Alghawli, A. S. (2019). Entropy analysis method for attacks detection. In 2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T) (pp. 443–446). IEEE. <https://doi.org/10.1109/PICST47496.2019.9061451>
2. Zaika, P. V. (2026). Study of methods for ensuring password security based on the analysis of actual attack vectors [Bachelor's qualification thesis, Kharkiv National University of Radio Electronics]. Electronic archive of NURE (OpenHUB).
3. Kanta, A., Coray, S., Coisel, I., & Scanlon, M. (2021). How viable is password cracking in digital forensic investigation? Analyzing the guessability of over 3.9 billion real-world accounts. *Forensic Science International: Digital Investigation*, 37, 301186. <https://doi.org/10.1016/j.fsidi.2021.301186>



4. Rodrigues, G. A. P., Fernandes, P. A. G., Serrano, A. L. M., Ribeiro, P. H. C., & de Oliveira, A. L. I. (2025). From RockYou to RockYou2024: Analyzing password patterns across generations, their use in industrial systems and vulnerability to password guessing attacks. *Journal of Internet Services and Applications*, 16(1), 69–86. <https://doi.org/10.5753/jisa.2025.4384>
5. Nisenoff, A., Golla, M., Wei, M., Hainline, J., Szymanek, H., Braun, A., Hildebrandt, A., Christensen, B., Langenberg, D., and Ur, B. (2023). A two-decade retrospective analysis of a university's vulnerability to attacks exploiting reused passwords. In *Proceedings of the 32nd USENIX Conference on Security Symposium (SEC '23)*. USENIX Association, USA, 287, pp. 5127–5144.
6. Ou, W., Sun, C., Pang, M., Wu, Z., & Chen, J. (2026). Improving targeted password guessing attacks by using personally identifiable information and old password. *Cybersecurity* 9, 9. <https://doi.org/10.1186/s42400-025-00430-0>
7. Wang, D., Shan, X., Dong, Q., Cheng, H., & Wang, P. (2023). No single silver bullet: Measuring the accuracy of password strength meters. In *32nd USENIX Security Symposium (USENIX Security 23)* (pp. 947–964). USENIX Association.
8. National Institute of Standards and Technology. (2024). Digital identity guidelines: Authentication and lifecycle management (NIST Special Publication 800-63B-4). U.S. Department of Commerce. <https://doi.org/10.6028/NIST.SP.800-63b-4>
9. Radivilova, T., Kirichenko, L., Tawalbeh, M., & Ilkov, A. (2021). Detection of anomalies in telecommunication traffic by statistical methods. *Electronic professional scientific publication "Cybersecurity: Education, Science, Technology"*, 3(11), 183–194. <https://doi.org/10.28925/2663-4023.2021.11.183194>
10. Jiang, J., Zhou, A., Liu, L., & Zhang, L. (2022). OMECDN: A password-generation model based on an ordered Markov enumerator and critic discriminant network. *Applied Sciences*, 12(23), 12379. <https://doi.org/10.3390/app122312379>
11. Hitaj, B., Gasti, P., Ateniese, G., Perez-Cruz, F. (2019). PassGAN: A Deep Learning Approach for Password Guessing. In: Deng, R., Gauthier-Umaña, V., Ochoa, M., Yung, M. (eds) *Applied Cryptography and Network Security. ACNS 2019. Lecture Notes in Computer Science()*, vol 11464. Springer, Cham. https://doi.org/10.1007/978-3-030-21568-2_11
12. Rando, J., Perez-Cruz, F., Hitaj, B. (2024). PassGPT: Password Modeling and (Guided) Generation with Large Language Models. In: Tsudik, G., Conti, M., Liang, K., Smaragdakis, G. (eds) *Computer Security – ESORICS 2023. ESORICS 2023. Lecture Notes in Computer Science*, vol 14347. Springer, Cham. https://doi.org/10.1007/978-3-031-51482-1_9
13. Imhof, G., Kanta, A., & Scanlon, M. (2025). Context-based password cracking dictionary expansion using generative pre-trained transformers. In *2024 Cyber Research Conference - Ireland, Cyber-RCI 2024 Institute of Electrical and Electronics Engineers Inc..* <https://doi.org/10.1109/Cyber-RCI60769.2024.10939663>
14. Wang, D., Zhang, Z., Wang, P., Yan, J., & Huang, X. (2016). Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS '16)* (pp. 1242–1254). Association for Computing Machinery. <https://doi.org/10.1145/2976749.2978339>
15. Radivilova, T., Kirichenko, L., Panteliev, V., Mazepa, A., & Bilodid, V. (2024). Analysis of authentication methods for full-stack applications and implementation of a web application with an integrated authentication system. *Innovative technologies and scientific solutions for industries*, 3 (29), 76–90. <https://doi.org/10.30837/2522-9818.2024.3.076>

Отримано редакцією журналу / Received: 29.08.26

Прорецензовано / Revised: 16.09.26

Схвалено до друку / Accepted: 26.10.26



This work is licensed under Creative Commons Attribution-noncommercial-sharealike 4.0 International License.