# Основи інформаційної та кібербезпеки

# Лекція 4.
# Програмно-технічний захист інформаційних систем

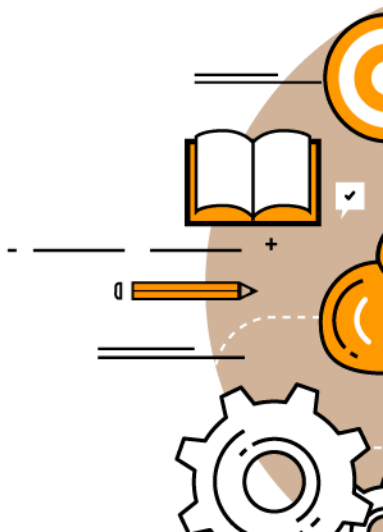
1. **Принципи архітектурної безпеки**
2. **Електронна ідентифікація користувачів**
3. **Використання програмних приманок**
4. **Прикладні аспекти захисту аутентифікаційних даних**

Центральним для програмно-технічного рівня є поняття ***сервісу безпеки.***

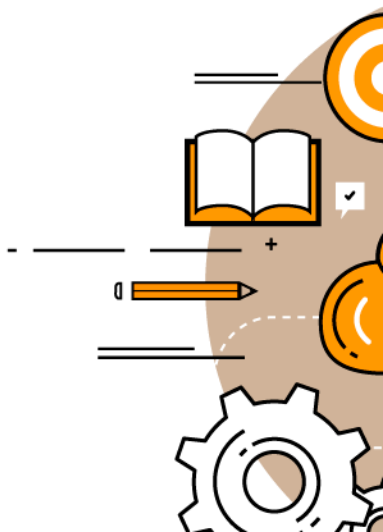
Визначення потрібного набору сервісів безпеки здійснюється виходячи з таких видів заходів безпеки:

❖**Засоби захисту від несанкціонованого доступу (НСД):**

- Засоби авторизації;
- Мандатне управління доступом;
- Виборче управління доступом;
- Управління доступом на основі ролей;
- Журналювання (Аудит).



- **Системи аналізу** та моделювання інформаційних потоків (CASE-системи).
- **Системи моніторингу мереж**:
  - Системи виявлення й запобігання вторгнень (IDS / IPS).
  - Системи запобігання витоків конфіденційної інформації (DLP-системи).
- **Аналізатори протоколів.**
- **Антивірусні засоби.**
- **Міжмережеві екрани.**
- **Криптографічні засоби:**
  - Шифрування;
  - Цифровий підпис.

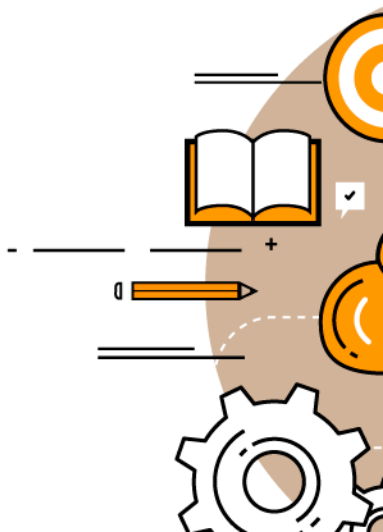


❖**Системи резервування**

- Резервне копіювання
- Відмовостійкий кластер
- Резервний Центр Обробки Даних (ЦОД) для катастрофостійкої ІС

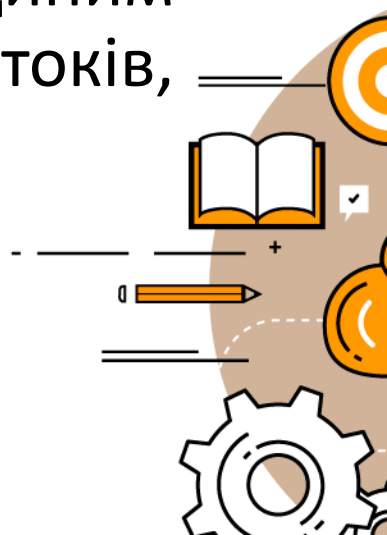
❖**Системи аутентифікації** на основі:

- Пароля;
- Ключа доступу (фізичного або електронного);
- Сертифікату;
- Біометричних даних.



# 1 Принципи архітектурної безпеки

- створення та впровадження єдиної політики безпеки;
- забезпечення конфіденційності і цілісності при мережевих взаємодіях;
- формування складених сервісів таким чином, щоб кожен компонент мав повний набір захисних засобів і, із зовнішньої точки зору, був єдиним цілим (не повинно бути інформаційних потоків, які йдуть до незахищених сервісів).



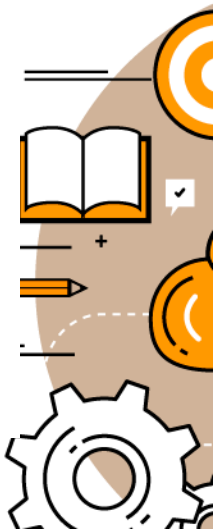
# 1 Архітектура ІС

***За ступенем розподіленості:***

- **настільні** (desktop), або локальні ІС, в яких всі компоненти (БД, СУБД, клієнтські додатки) знаходяться на одному комп'ютері;
- **розподілені** (distributed) ІС, в яких компоненти розподілені по декількох комп'ютерів.

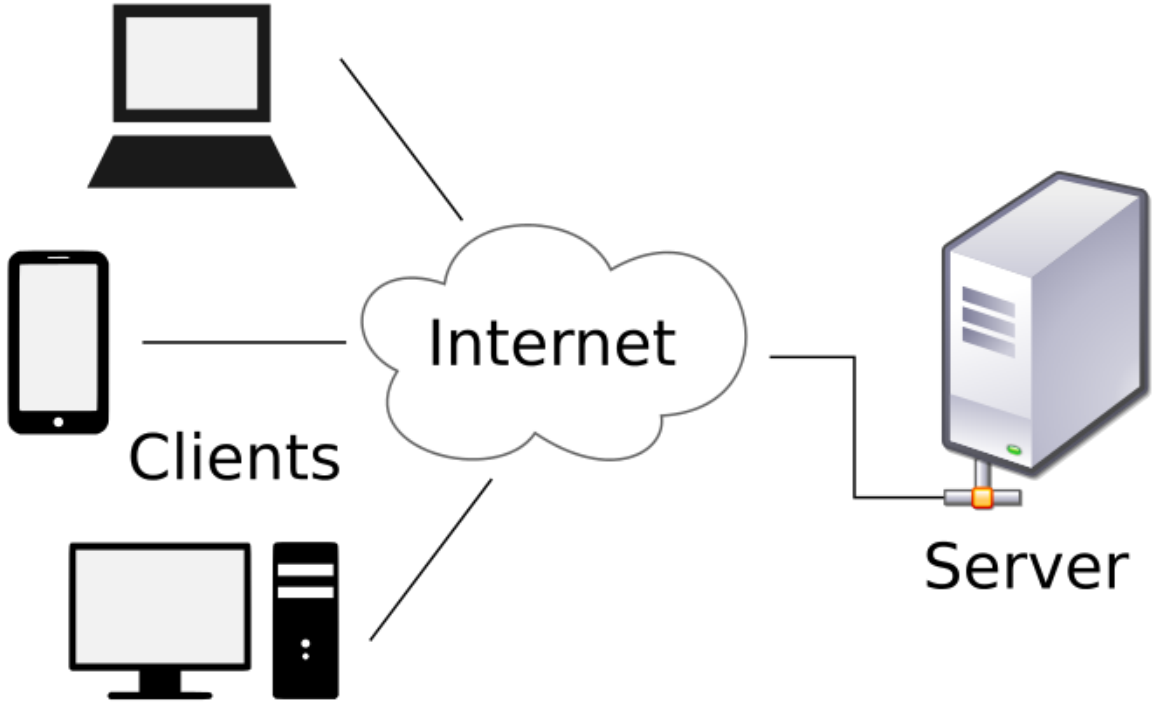
**Розподілені ІС:**

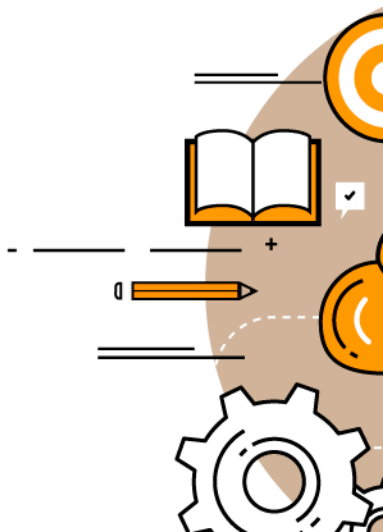
- *файл-серверні* ІС (база даних знаходиться на файловому сервері, а СУБД і клієнтські додатки знаходяться на робочих станціях.)
- *клієнт-серверні* ІС (база даних і СУБД знаходяться на сервері, а на робочих станціях знаходяться клієнтські додатки)



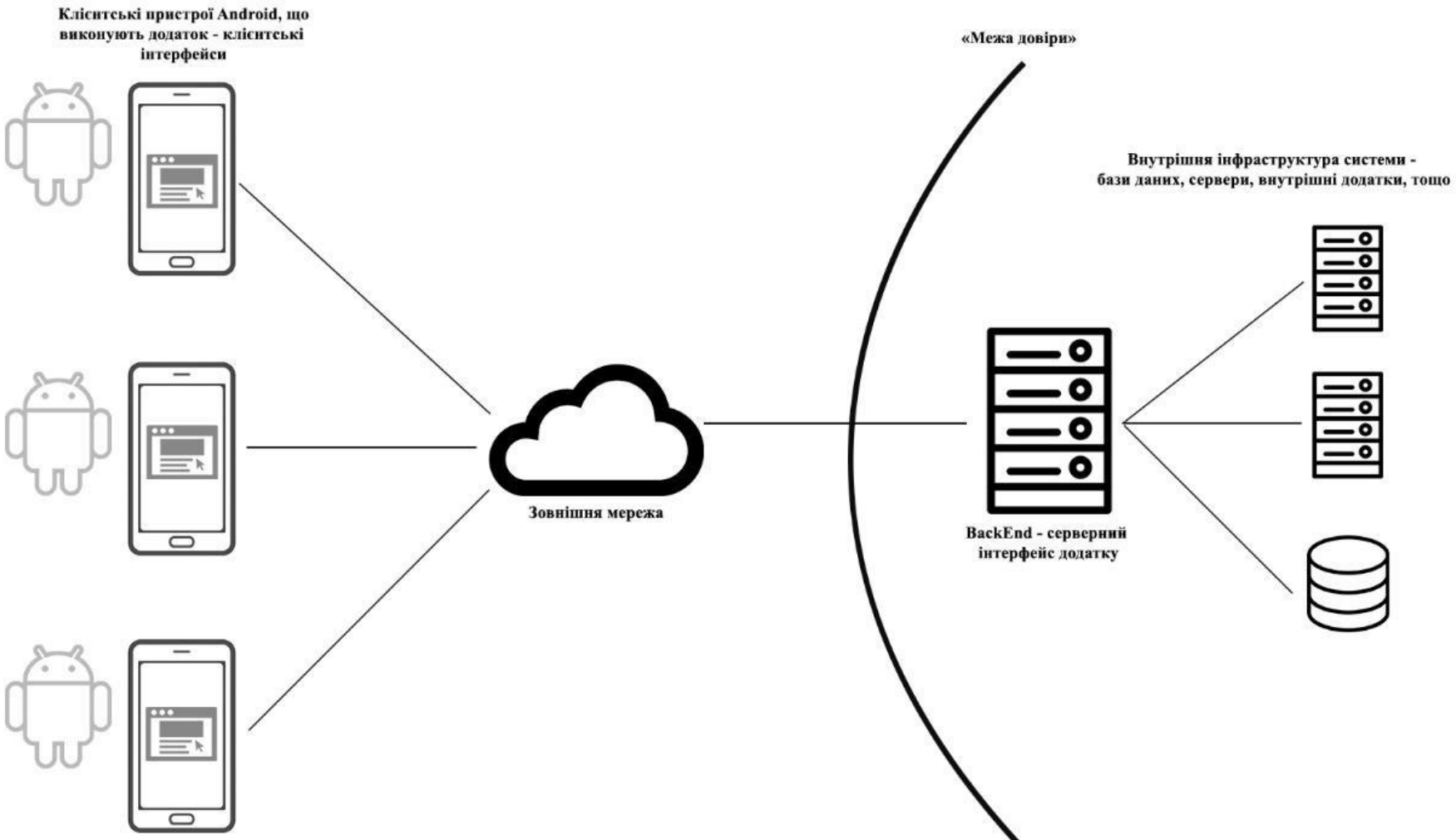
# 1 Архітектура ІС

## Клієнт-серверні ІС



# 1 Концепція безпеки ІС на Android




**Типова структура інформаційної системи клієнт-серверної архітектури з Android-додатками в якості клієнтів**

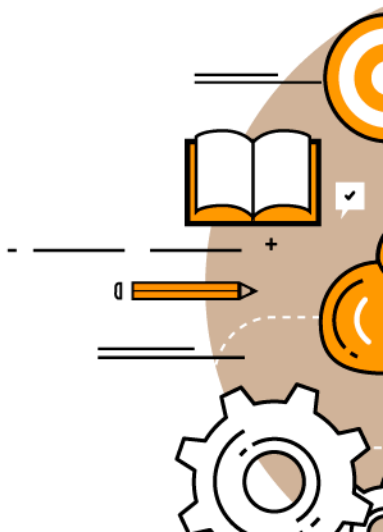
# 1 Концепція безпеки ІС на Android

ІС на Android:

- додаток для ОС Android в якості клієнтського інтерфейсу,
- WEB-сервер, що представляє собою інтерфейс Backend,
- інформаційна інфраструктура в рамках backend та комунікаційний канал за протоколом HTTP.

**Вектори атак:**

1) На процеси серверної частини;
2) На клієнтський додаток;
3) На канал комунікації.

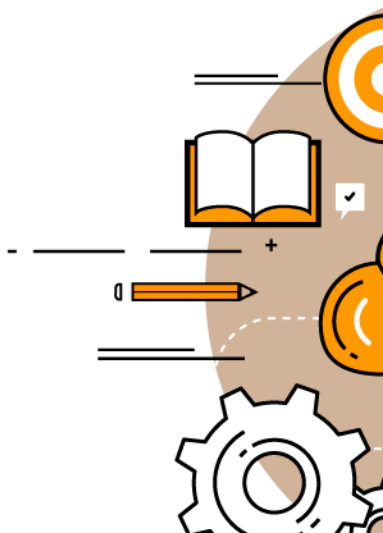


# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

**сервіси:**

- Аутентифікація, авторизація та розмежування доступу;
- Контроль сеансів;
- Бізнес-логіка;
- Валідація даних;
- Обробка помилок;



# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

**сервіси:**

- *Аутентифікація, авторизація та розмежування доступу* (експлуатація вразливостей механізмів авторизації та аутентифікації може призвести до горизонтального або вертикального підвищення привілеїв);

Вразливості: SQL ін’єкція у авторизаційному механізмі у випадку звернення до БД, відсутність обмежень на неуспішні спроби аутентифікації (захист від brute-force атак) тощо;

вразливості розмежування доступу: **IDOR** (Insecure Direct Object Reference – небезпечний прямий доступ до об’єкту), відсутність або некоректна перевірка токену доступу тощо.

# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

- *Контроль сеансів* (експлуатація вразливостей пов’язаних з процесами авторизації, оскільки оперують переважно одною множиною ресурсів системи);

Вразливості:

- відсутність перевірки ідентифікатора сесії при аутентифікації (можливість реалізації атаки **Session Fixation**),
- використання ідентифікатору сесій низької складності, тобто таких, що можуть бути підібраними за відносно короткий час,
- відсутність прив’язки сесії до пристрою та відсутність (або недостатність) обмеженості часу дійсності сеансу.

# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

- *Бізнес-логіка* (не мають чіткої класифікації, різнорідні проблеми в реалізації тих чи інших алгоритмів у роботі з даними в додатку, тому залежать від конкретної реалізації);

Вразливості дозволяють використовувати незапланований при проектуванні функціонал, не порушуючи коректність роботи програми:

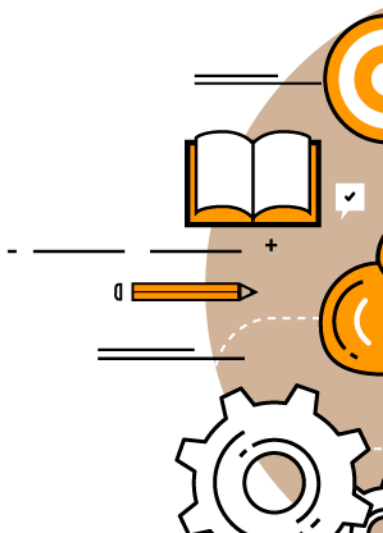
- недостатня валідація порядку дій у процесі, порушення в обробках паралельних процесів (**Race condition**).

# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

- *Валідація даних* (обробка даних, що отримуються з недовіреного середовища - можуть мати необмежений вплив на систему, що є критичним при обробці даних з обмеженим доступом);

**Вразливості**: усі типи атак ін’єкцій (SQL, OS-command, XML, JavaScript тощо).

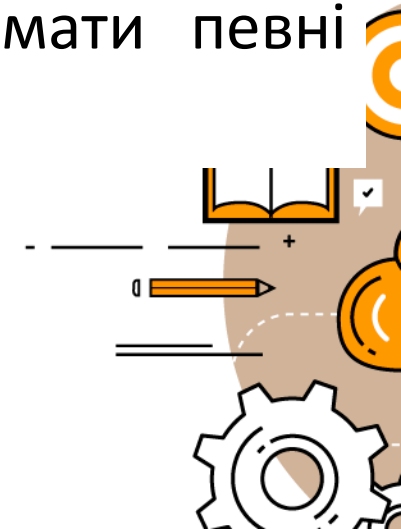


# 1 Концепція безпеки ІС на Android

## 1. Вектори атак на серверну частину ІС:

- *Обробка помилок* (стабілізація роботи системи у випадку нештатної ситуації, яка, зокрема, може бути створена штучно).

Недостатня або некоректна обробка помилок може дозволити порушнику змінити коректний хід роботи програми та, таким чином, отримати доступ до розголошенної технічної інформації або отримати певні можливості впливу на систему.



# 1 Концепція безпеки ІС на Android

## 2. Вектори атак на клієнтський додаток:

Атаки, що здійснюються на клієнтський мобільний додаток, можуть мати вплив у першу чергу на ті дані, що обробляються локально.

**сервіси:**

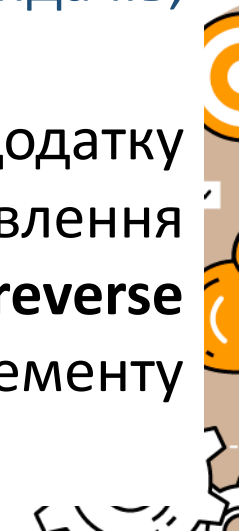
- *Інтерфейс взаємодії з користувачем;*
- *Міжпроцесна взаємодія;*
- *Локальне накопичення даних;*

# 1 Концепція безпеки ІС на Android

## 2. Вектори атак на клієнтський додаток:

- *Інтерфейс взаємодії з користувачем (*в системі Android забезпечується за допомогою елементів **Activity** та **WebView**. Activity є абстракцією робочого екрану додатку з елементами інтерфейсу, призначеними для взаємодії з користувачем. Кожен елемент Activity надає користувачу інтерфейс для взаємодії з даними системи. Компоненти **WebView** призначені для відображення веб-сторінок і в цілому аналогічні іншим типам веб-переглядачів, а отже й наслідують більшість їхніх вразливостей*);*

**Вразливості**: обумовлені закладеними в код додатку елементами конфіденційної інформації – ризик їх виявлення шляхом аналізу за методами зворотної розробки (**reverse engineering**). Можливості програмного доступу до елементу інтерфейсу з боку інших додатків, що функціонують в ОС.



# 1 Концепція безпеки ІС на Android

## 2. Вектори атак на клієнтський додаток:

- *Міжпроцесна взаємодія (*реалізується переважно за допомогою функціоналу системного віртуального пристрою **Binder**. Обмін даними виконується за допомогою механізмів **BroadcastReceiver** та **ContentProvider**, що, відповідно, отримують та надають дані іншим процесам*);*

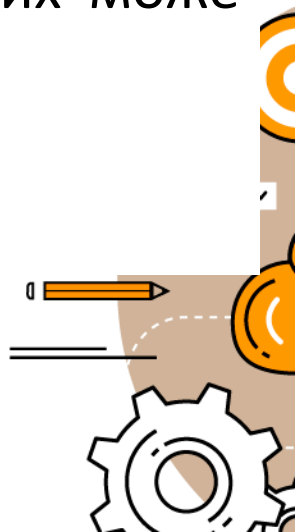
**Вразливості**: пов`язані, здебільшого, з валідацією або логікою додатку. Так, наприклад, вразливість в реалізації **ContentProvider** може створювати витоки інформації до інших додатків, а в **BroadcastReceiver** – непередбачені можливості для функціоналу.

# 1 Концепція безпеки ІС на Android

## 2. Вектори атак на клієнтський додаток :

- *Локальне накопичення даних (*в додатках ОС Android може здійснюватися за допомогою файлів у локальній файловій системі (зокрема спеціальних – Shared Preferences), за допомогою локальних баз даних, таких як SQLite, або в системних сховищах, таких як KeyChain*);*

**Вразливості**: збереження даних додатком на SD карті – архітектура ОС Android надає менші гарантії безпеки даних збережених на зовнішньому носії, а отже доступ до них може бути отриманий з інших процесів.



# 1 Концепція безпеки ІС на Android

## 3. Вектори атак на канал комунікації:

*Використання протоколу HTTP без додаткового захисту саме по собі є серйозною вразливістю системи, оскільки при цьому відсутні будь-які можливості для забезпечення конфіденційності та цілісності інформації при передачі. Для забезпечення цих характеристик використовується захищений протокол з'єднання – HTTPS.*

Мобільні пристрої працюють, здебільшого, у недовіреному середовищі, яке може бути контрольоване потенційним порушником, можливості для використання звичайного протоколу HTTPS є обмеженими – порушник може контролювати DNS сервер мережі та перехоплювати клієнтський трафік.

Для подолання цієї проблеми використовується технологія **SSL Pinning .**

# 1 Концепція безпеки ІС на Android

## 3. Вектори атак на канал комунікації:

- **MitM атака** (атака **«людина посередині»**) на незахищений канал комунікації (HTTP);
- **MitM атака з підміною DNS** серверу в мережі на захищений канал комунікації (HTTPS);
- Експлуатація вразливостей імплементації технології SSL Pinning для здійснення атаки на захищений канал;
- Криптографічні атаки на захищений канал комунікації

# 1 Концепція безпеки ІС на Android

## Моделі порушника:

A. **Порушник є стороннім атакуючим**, що не має доступу до конкретного авторизованого клієнтського додатку, доступу до клієнтського пристрою та доступу до серверної частини системи.
B. **Порушник має певний рівень доступу** до пристрою клієнта, що є прихованим від самого клієнта (**троян**), доступ до серверної частини та безпосередньо до додатку відсутній.
C. **Порушник представляє собою клієнта в системі** та може змінювати клієнтський додаток довільним чином. Доступ до серверної частини системи – на рівні звичайного клієнта.
D. **Порушник має повний фізичний доступ** до пристрою клієнта (дії поза інформаційним простором (викрадення, конфіскація,тощо) .

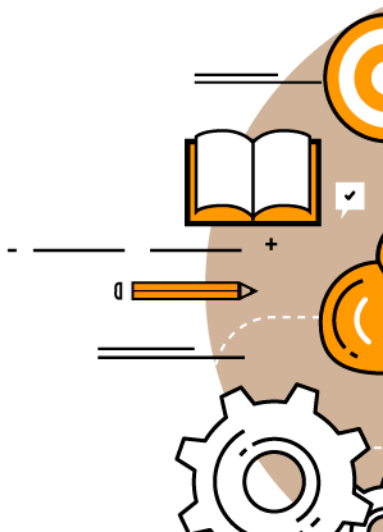
# 1 Концепція безпеки ІС на Android

**Порівняння поверхні атаки для різних моделей порушника**

| Модель | Backend | Додаток | Канал комунікації |
|---|---|---|---|
| **A** | Частковий | Відсутній | Повний/частковий |
| **B** | Частковий | Частковий/Повний | Відсутній |
| **C** | Частковий | Повний | Н/в |
| **D** | Частковий | Повний | Н/в |

В таблиці вказуються наступні можливі типи впливу:

- Н/в – вплив можливий, але не має сенсу для даної моделі порушника;
- Відсутній – вплив неможливий;
- Частковий – порушник має обмежені можливості щодо атаки на об’єкт;
- Повний – порушник не має технічних обмежень середовища щодо атаки на об’єкт.



# 2 Електронна ідентифікація користувачів

**Ідентифікація та автентифікація** –
основа програмно-технічних засобів безпеки, оскільки решта сервісів розраховані на обслуговування іменованих суб’єктів.

**Ідентифікація** дозволяє суб’єкту (користувачу, процесу, що діє від імені певного користувача, або іншому апаратно-програмному компоненту) назвати себе (повідомити своє ім’я).
За допомогою **автентифікації** друга сторона переконується, що суб’єкт дійсно той, за кого він себе видає.
Автентифікація може бути ***односторонньою*** (звичайно клієнт доводить свою достовірність серверу) і ***двосторонньою*** (взаємною).

# 2 Електронна ідентифікація користувачів

***Щоб визначити чиюсь справжність, можна скористатися трьома факторами:***

| | | |
|---|---|---|
| ***Пароль** - то, що ми знаємо (слово, PIN-код, код для замка, графічний ключ)* | ***Пристрій (токен)** - то, що ми маємо (пластикова карта, ключ від замка, USB-ключ)* | ***Біометрика** - то, що є частиною нас (відбиток пальця, портрет, сітківка ока)* |
|  |  |  |

# 2 Електронна ідентифікація користувачів

Значно підвищити надійність **парольного захисту** дозволяють такі заходи:

- накладення технічних обмежень (пароль повинен бути не дуже коротким, та містити букви, цифри, знаки пунктуації тощо);
- управління терміном дії паролів, їх періодична зміна;
- обмеження доступу до файлу паролів;
- обмеження кількості невдалих спроб входу в систему;
- навчання користувачів;
- використання програмних генераторів паролів.

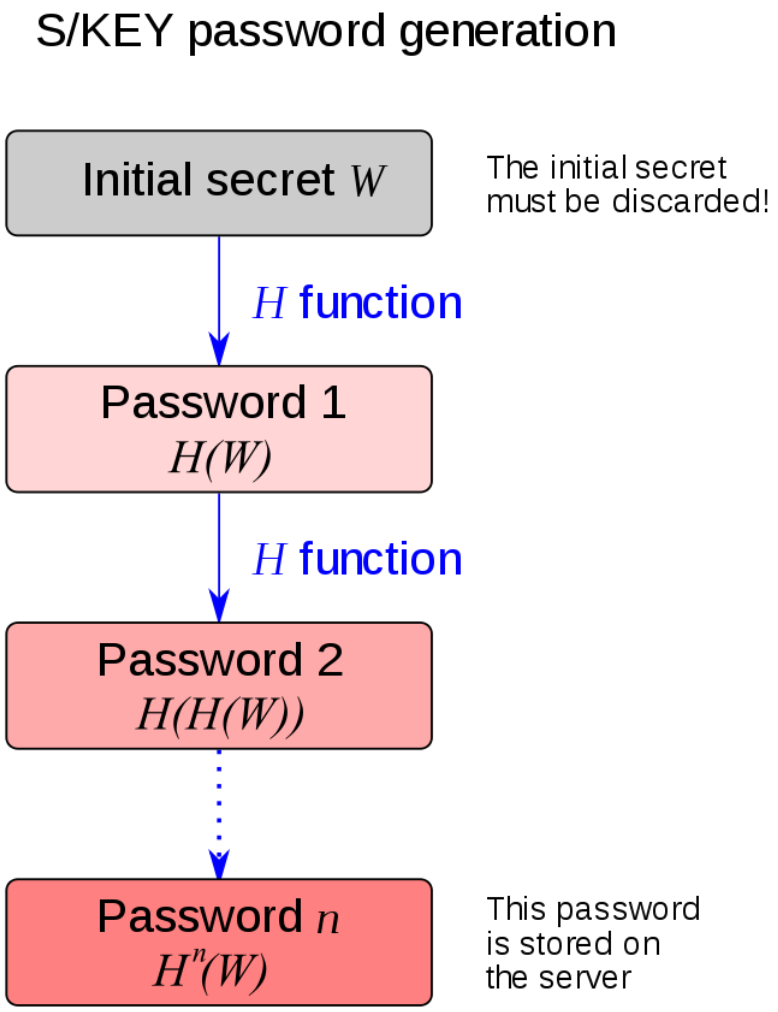
# 2 Електронна ідентифікація користувачів

**Одноразові паролі**:

- система **S/KEY** компанії Bellcore

За рахунок використання односторонньої функції, відомої користувачу та серверу, та секретного ключа, відомого тільки користувачу, перехоплення пароля, рівно як і отримання доступу до сервера автентифікації, не дозволяють дізнатися секретний ключ і передбачити наступний одноразовий пароль.

Система S/KEY має статус Internet-стандарту (RFC 1938).

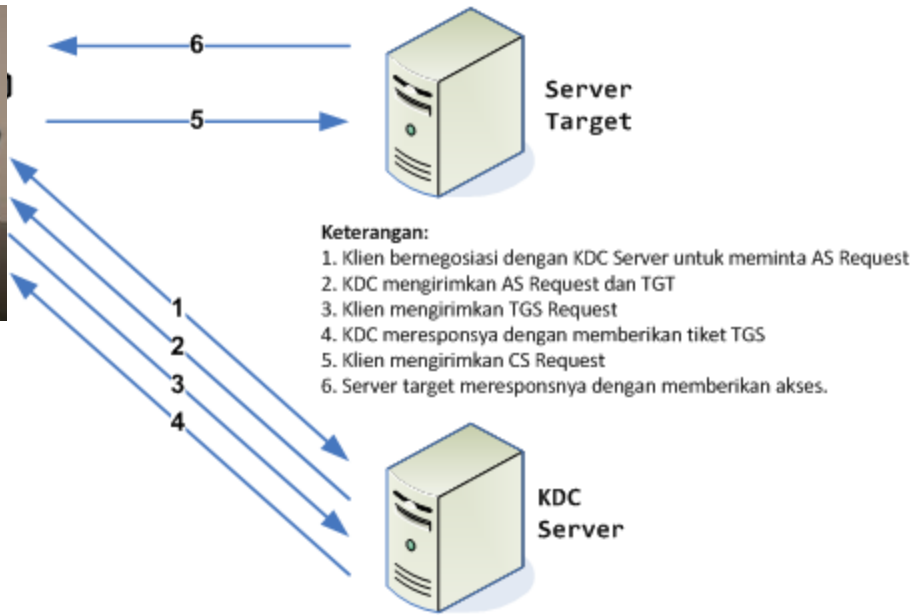


# 2 Електронна ідентифікація користувачів

## Протокол Керберос (Kerberos)

Схема передбачає взаємодію між трьома програмними компонентами

- клієнтом C,
- сервером Kerberos
- прикладним сервером S.

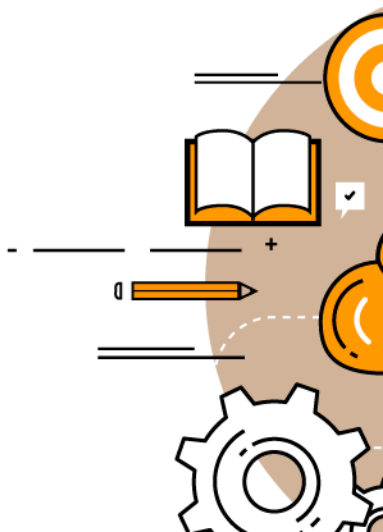



**Keterangan:**
1. Klien bernegosiasi dengan KDC Server untuk meminta AS Request
2. KDC mengirimkan AS Request dan TGT
3. Klien mengirimkan TGS Request
4. KDC meresponsya dengan memberikan tiket TGS
5. Klien mengirimkan CS Request
6. Server target meresponsnya dengan memberikan akses.

# 2 Електронна ідентифікація користувачів

Запис кожного користувача і кожного прикладного сервера в базі даних Kerberos містить наступні компоненти

- Ідентифікатор суб'єкта;
- Секретний ключ суб'єкта;
- Дату закінчення терміну дії секретного ключа;
- Максимальний термін життя дозволів, які видаються суб'єкту;
- Номер версії секретного ключа суб'єкта;
- Дату останньої модифікації запису;
- Іншу службову інформацію.



# 2 Електронна ідентифікація користувачів

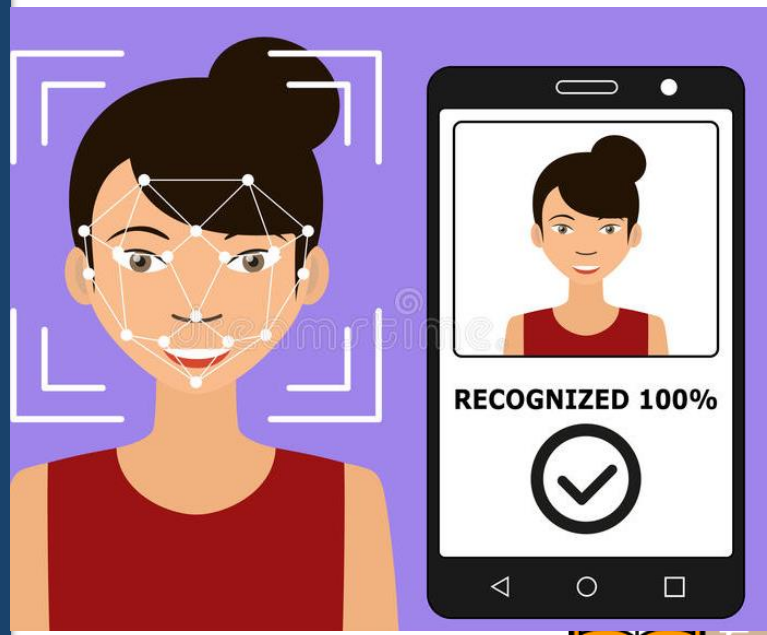
Система **Kerberos** є довіреною третьою стороною (тобто стороною, якій довіряють все), що володіє секретними ключами обслуговуваних суб'єктів і допомагає їм у попарній перевірці достовірності.

- Щоб дістати доступ до **S**, **C** посилає Kerberos запит, який містить відомості про нього і про запрошувану послугу.
- У відповідь Kerberos повертає так званий “квиток”, зашифрований секретним ключем сервера, і копію частини інформації з “квитка”, зашифровану секретним ключем клієнта.
- Клієнт повинен розшифрувати другу порцію даних і переслати її разом з “квитком” серверу. Сервер, розшифрувавши “квиток”, може порівняти його вміст з додатковою інформацією, присланою клієнтом. Збіг свідчить про те, що клієнт зміг розшифрувати призначені йому дані, тобто продемонстрував знання секретного ключа. Отже, клієнт – саме той, за кого себе видає.

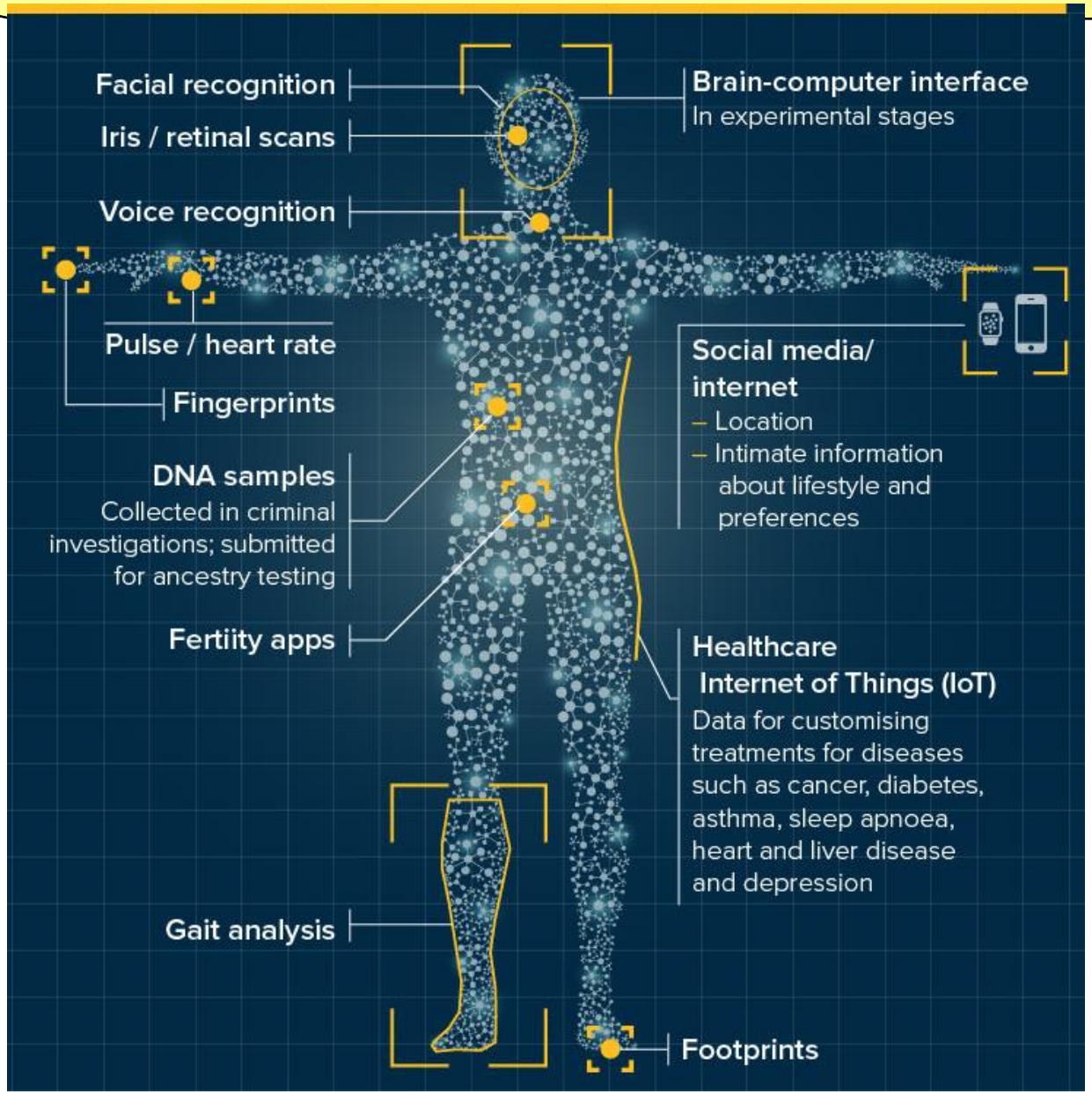
# 2 Електронна ідентифікація користувачів

## Біометрична аутентифікація

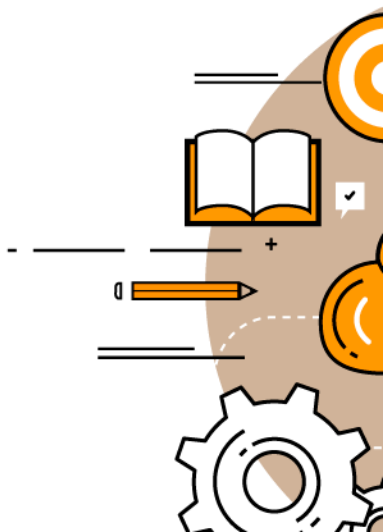
***Системи біометричного захисту використовують унікальні для кожної людини вимірювані характеристики для перевірки особи індивіда. Біометричний захист ефективніший ніж такі методи як, використання смарт-карток, паролів, PIN-кодів.***



# 2 Біометрична аутентифікація

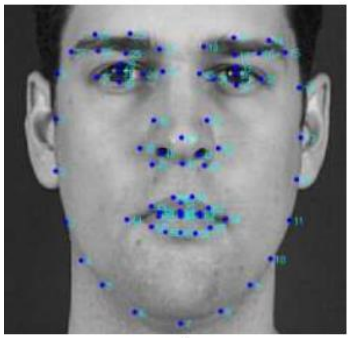



**Тіло людини, як найбільша платформа даних**

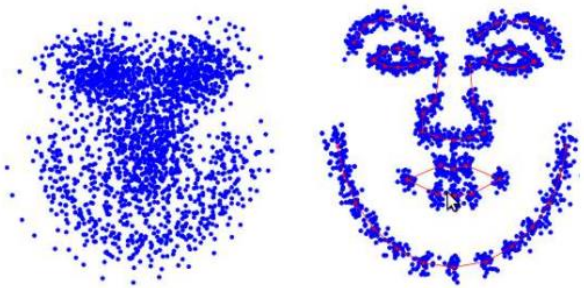


# 2 Біометрична аутентифікація

# 2 Статичні методи ідентифікації

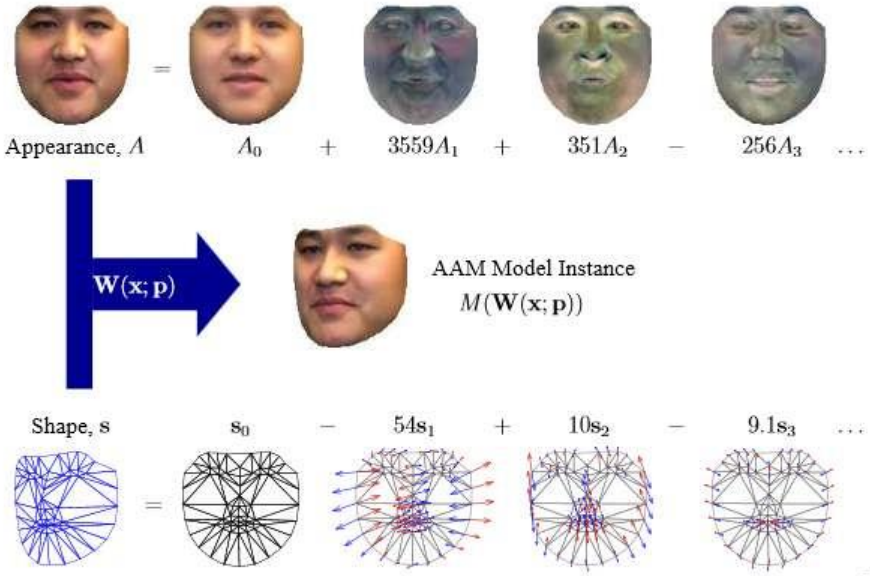


а)



б)

Рис.1. Метод активних моделей зовнішнього вигляду обличчя :
а) приклад розмітки форми обличчя;
б) координати точок форми обличчя до і після нормування



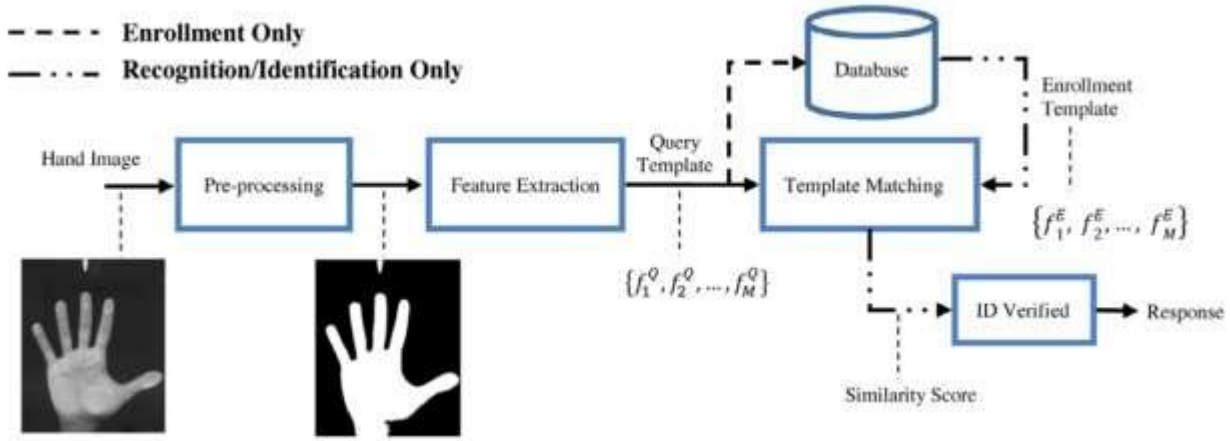
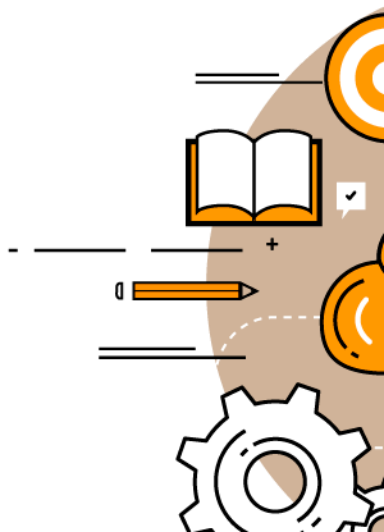




Рис.2. Структурна схема біометричного розпізнавання геометрії руки



# 2 Статичні методи ідентифікації

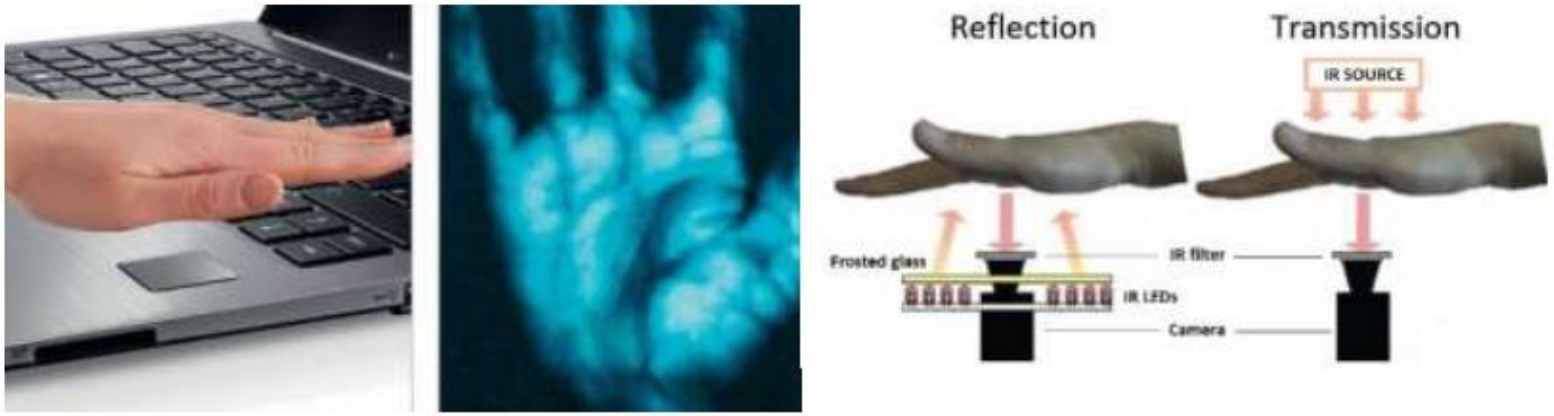



Рис.3. Ідентифікація особистості за візерунком вен руки

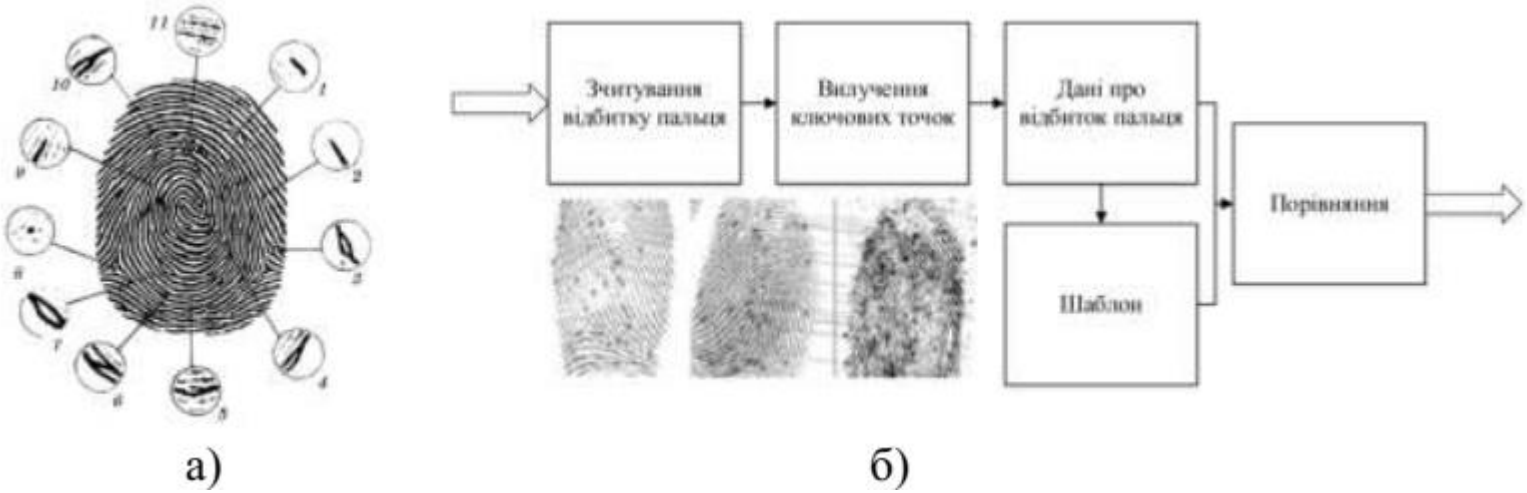



Рис.4. Метод ідентифікації за відбитком пальця:
а) основні типи контрольних параметрів відбитку пальця ;
б) структурна схема дактилоскопічного розпізнавання .

# 2 Статичні методи ідентифікації

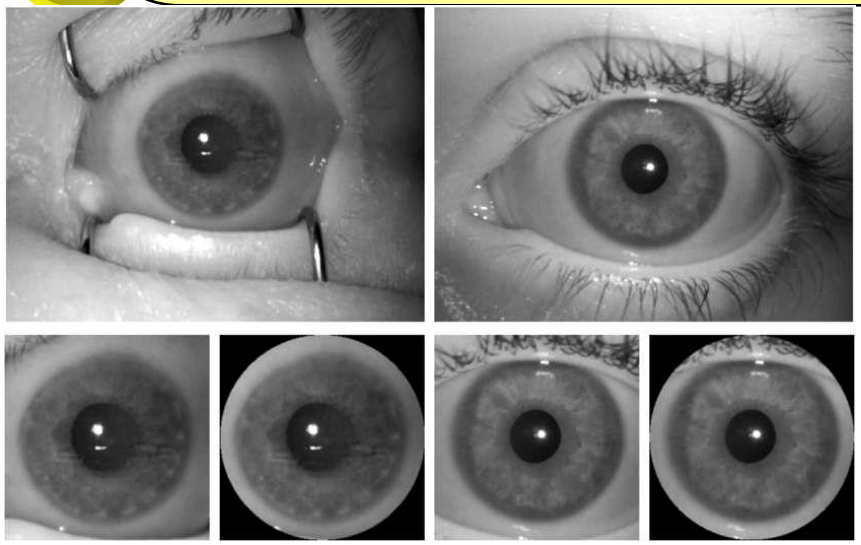


Рис.5. Інфрачервоне сканування рогівки ока

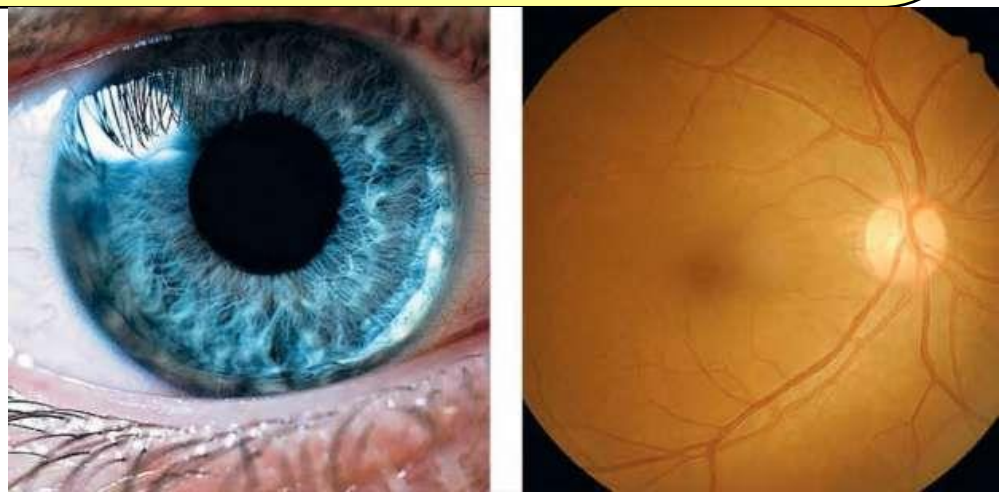


Рис.6. Метод ідентифікації за райдужною оболонкою ока

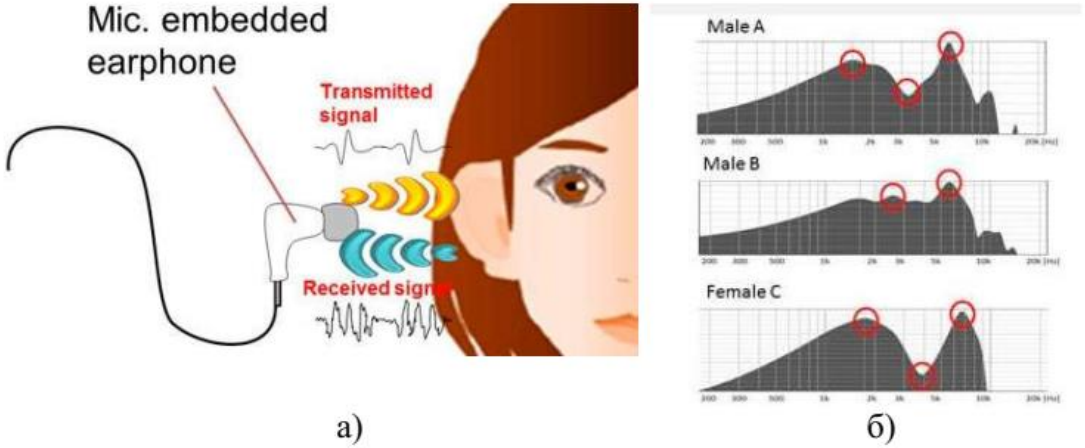



Рис. 7. Метод ідентифікації особи за резонуванням звуку в порожнині вуха :
а) принцип проведення методу; б) зображення звукової луни у різних людей

# 2 Статичні методи ідентифікації

## Показники ефективності для біометричних систем:

'Невірний коефіцієнт відповідності' (**FAR** = False Accept Rate): відношення числа разів, коли нелегальний користувач був пропущений в систему до загальної кількості спроб проникнути в систему;

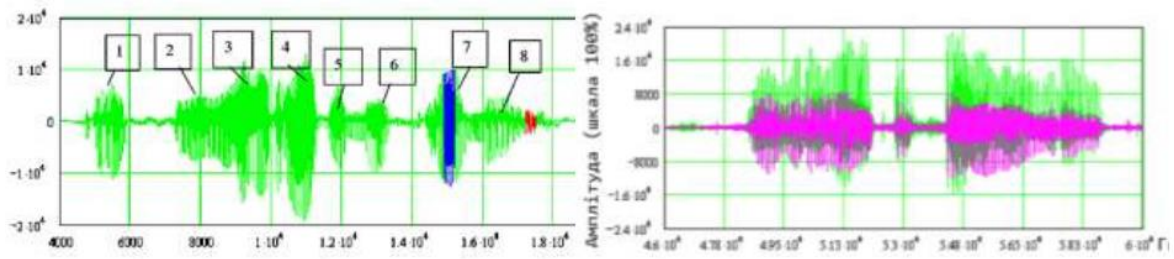
'False non-match rate' (FNMR, також називається **FRR** = False Reject Rate): відношення числа разів, коли легальний користувач був не пропущений в систему до загальної кількості спроб увійти в систему.

# 2 Статичні методи ідентифікації

## Порівняння статичних методів біометричної ідентифікації особистості

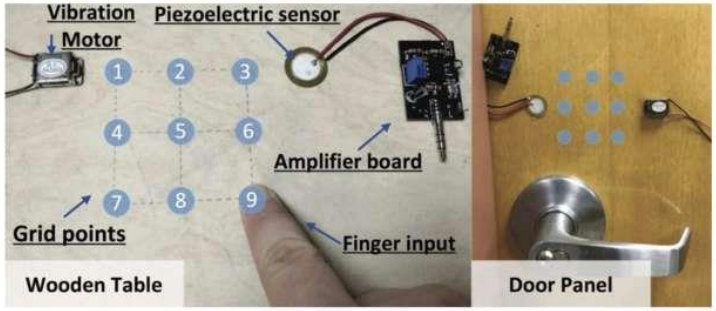
| Біометрична метод | FAR % | FRR % | Фальсифікація | Незмінність характеристик | Чутливість до зовн. факторів | Швидкість автентифікації | Вартість |
|---|---|---|---|---|---|---|---|
| **Відбиток пальця** | 0,001 | 0,6 | Можлива | Низька | Висока | Висока | Низька |
| **Розпізнавання форми лиця 2D** | 0,1 | 2,5 | Можлива | Низька | Висока | Середня | Середня |
| **Розпізнавання форми лиця 3D** | 0,0005 | 0,1 | Проблематична | Висока | Низька | Низька | Висока |
| **Райдужна оболонка ока** | 0,00001 | 0,016 | Безуспішна | Висока | Середня | Висока | Висока |
| **Сітківка ока** | 0,0001 | 0,4 | Неможлива | Середня | Висока | Низька | Висока |
| **Візерунок вен** | 0,0008 | 0,01 | Неможлива | Середня | Середня | Висока | Середня |
| **Розпізнавання за формою губ** | 0,2 | 3,8 | Можлива | Низька | Середня | Середня | Середня |
| **Розпізнавання за формою носа** | 0,8 | 4,9 | Можлива | Середня | Низька | Середня | Середня |
| **Розпізнавання за формою уха (резонування)** | 0,002 | 0,1 | Проблематична | Висока | Висока | Висока | Низька |
| **Розпізнавання за кодом ДНК** | 0,00001 | 0,0012 | Проблематична | Висока | Середня | Низька | Висока |
| **Розпізнавання за формою кисті** | 0,1 | 2,5 | Можлива | Низька | Середня | Середня | Середня |

# 2 Динамічні методи ідентифікації особистості

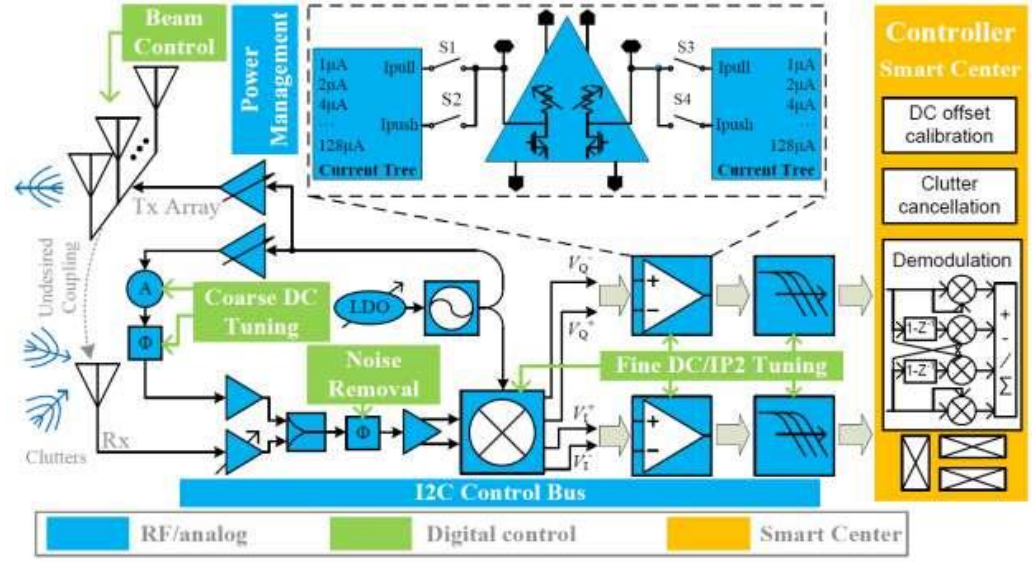



а) б)

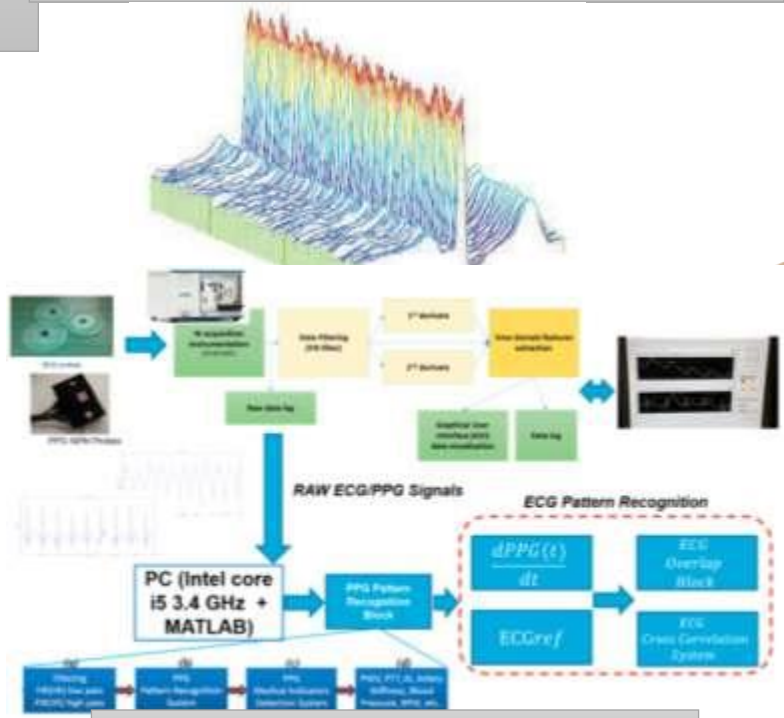
Запис фрагментів мови: а) голосова фраза з виділенням фрагментів коливання різних частот; б) мовна фраза проходження звуку через ключицю (рожевий колір) та шийну зони (зелений колір).




Принцип ідентифікації за вібрацією пальців.




Ідентифікація за геометрією серця за допомогою допплерівського радару




за сигналами роботи серця

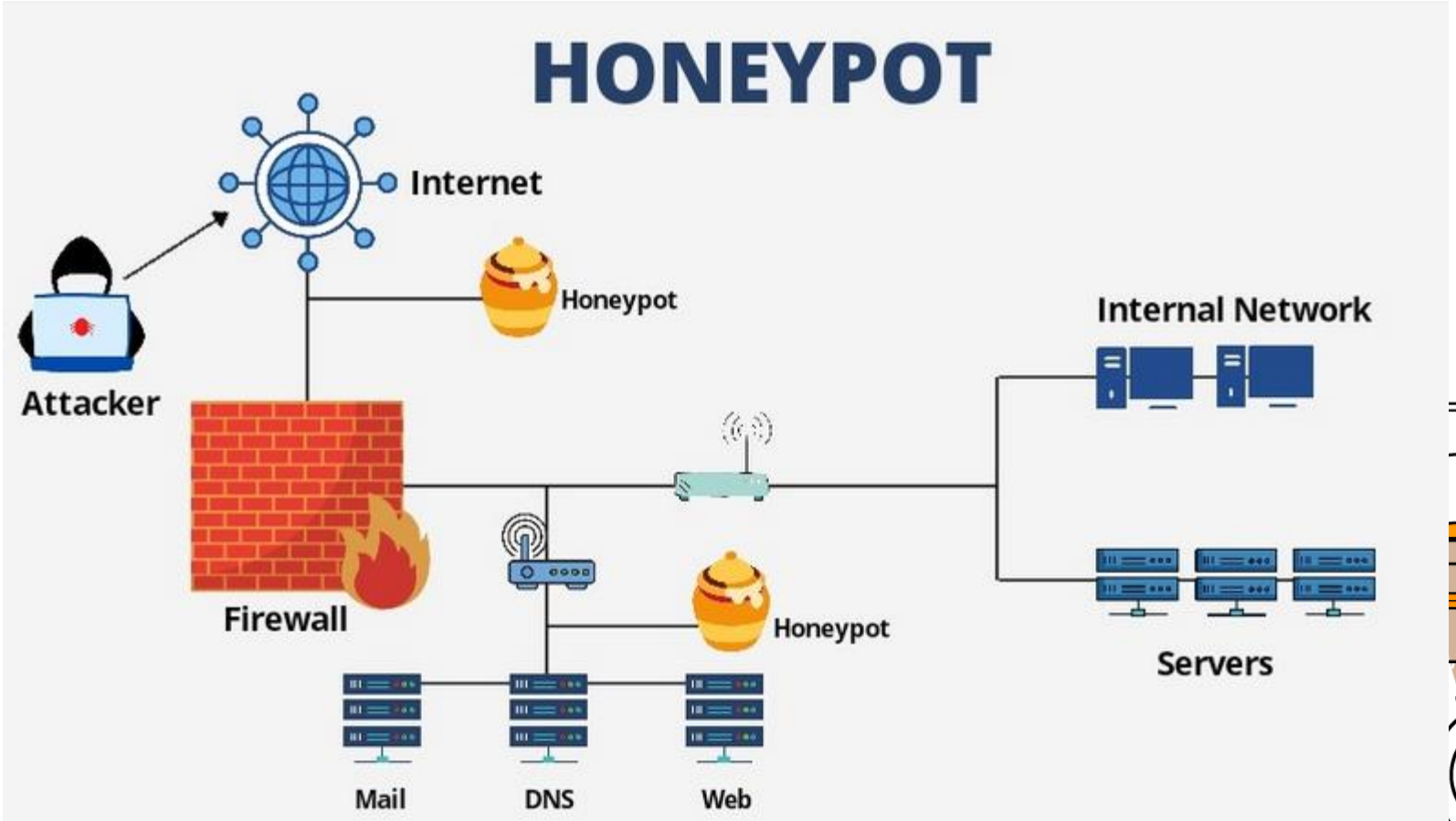
# 2 Динамічні методи ідентифікації особистості

## Порівняння динамічних методів біометричної ідентифікації

| Біометричний метод | FAR % | FRR % | Фальсифікація | Незмінність характеристик | Чутливість до зовн. факторів | Швидкість автентифікації | Вартість |
|---|---|---|---|---|---|---|---|
| **Розпізнавання за ходою людини** | 0,001 | 0,6 | Можлива | Низька | Висока | Висока | Низька |
| **Розпізнавання за запахом тіла** | 0,1 | 1,5 | Проблематична | Низька | Висока | Середня | Середня |
| **Розпізнавання за складом поту** | 0,0005 | 0,1 | Проблематична | Висока | Низька | Низька | Середня |
| **Розпізнавання за мікровібрацією пальців** | 0,00001 | 0,016 | Проблематична | Висока | Середня | Висока | Висока |
| **Розпізнавання за почерком (клавіатура)** | 0,012 | 0,4 | Можлива | Середня | Низька | Низька | Низька |
| **Розпізнавання за електроміограмою** | 0,0008 | 0,023 | Неможлива | Середня | Середня | Висока | Середня |
| **Розпізнавання за фотоплетизмограмою** | 0,0007 | 0,01 | Проблематична | Середня | Середня | Висока | Низька |
| **Розпізнавання за геометрією серця** | 0,003 | 0,012 | Неможлива | Середня | Середня | Висока | Середня |
| **Розпізнавання за електрокардіограмою** | 0,0005 | 0,01 | Неможлива | Середня | Середня | Висока | Середня |

# 3 Використання програмних приманок як засобу забезпечення ІБ

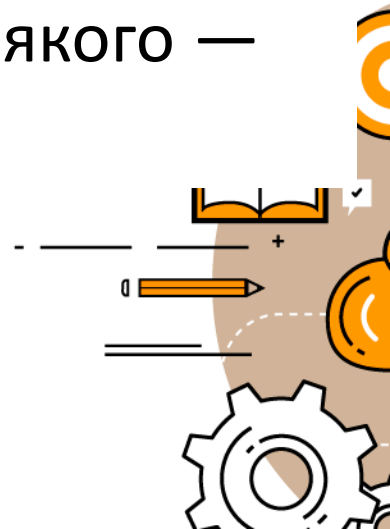
**Honeypot** («Пастка») (англ. *горщик з медом*) — ресурс, що є собою приманкою для зловмисників.



# 3 Використання програмних приманок як засобу забезпечення ІБ

**Мета Honeypot** — зазнати атаки або несанкціонованого дослідження, що згодом дозволить вивчити стратегію зловмисника та визначити перелік засобів, за допомогою яких можуть бути завдані удари реально наявним об'єктам безпеки.

Реалізація Honeypot може бути спеціальним виділеним сервером, або мережевим сервісом, завдання якого — привернути увагу зловмисника.



# 3 Використання програмних приманок

- **Honeypot** є ресурсом, що без будь-якого впливу на нього є неактивним. Honeypot збирає невелику кількість інформації, після аналізу якої будується статистика методів, якими користуються зловмисники, а також визначається наявність якихось нових рішень, які згодом будуть застосовуватися в боротьбі з ними.

Типи даних, які збирають приманки:

- Сполучення клавіш вводяться зловмисником
- IP-адреса зловмисника
- Імена користувачів і різні привілеї, які використовуються зловмисником.
- Дані, до яких зловмисник:
    - отримав доступ; видалив; змінив

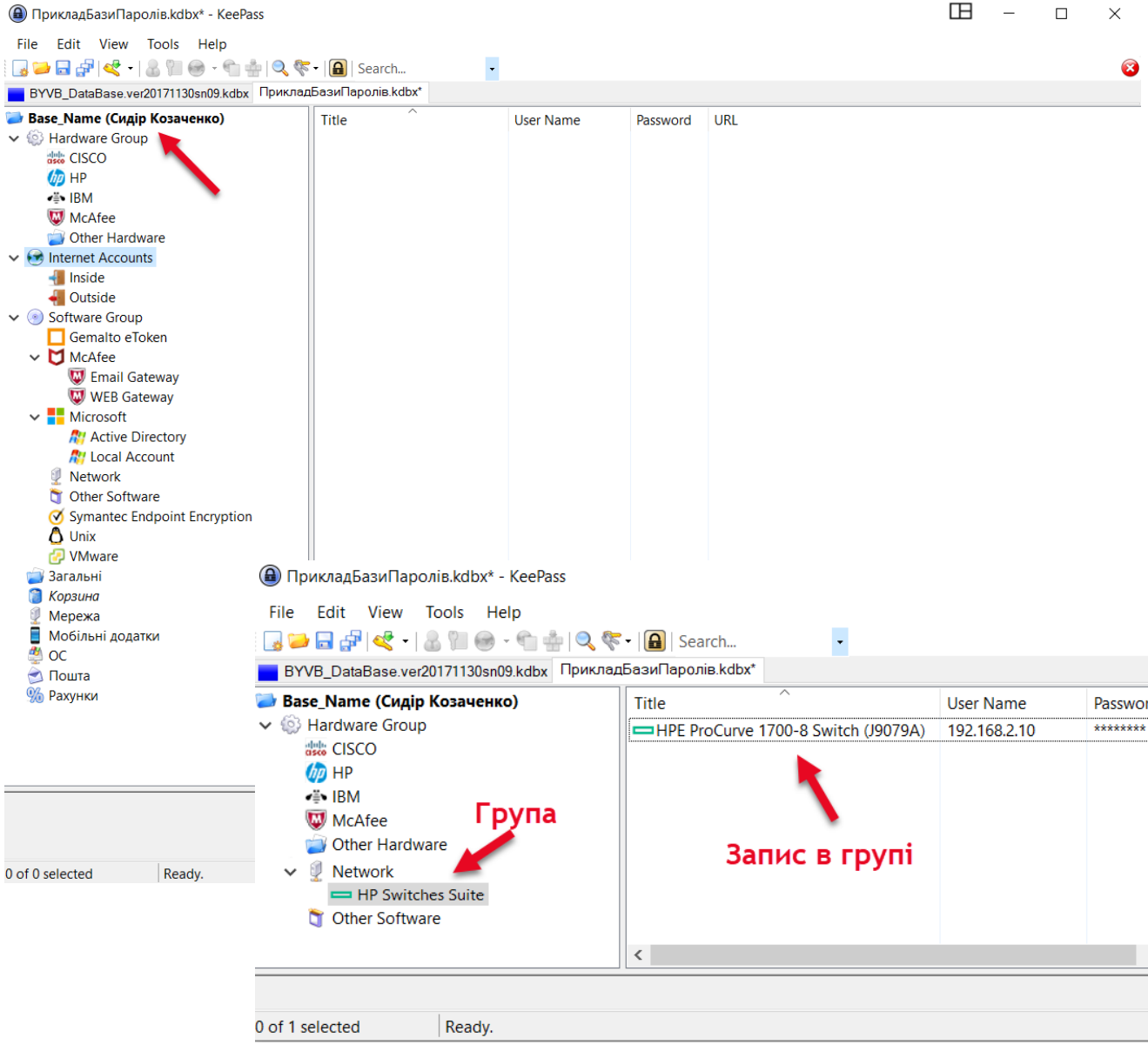
# 4 Прикладні аспекти захисту аутентифікаційних даних

В якості вирішення будемо використовувати менеджер паролів **KeePass** і флешку з апаратним шифрування від **SAFEXS**.

**KeePass Password Safe** - безкоштовна програма (менеджер паролів), яка дозволяє зберігати всі паролі користувача, використовуючи один головний майстер-пароль.

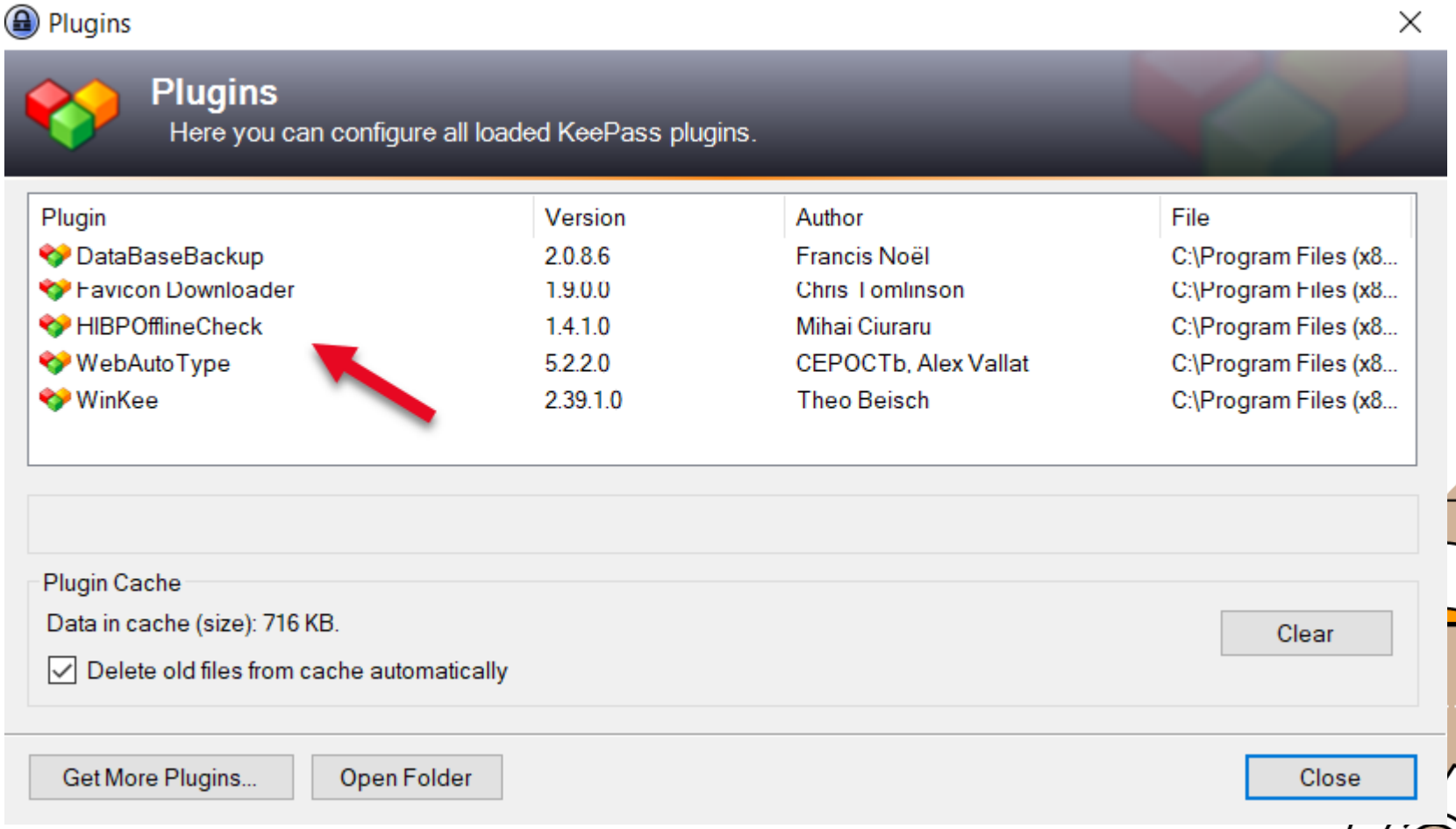
**KeePass** на сьогодні вважається однією з кращих, з відкритим кодом, безкоштовних і легких у використанні крос-платформних програм для зберігання паролів.

# 4 Прикладні аспекти захисту аутентифікаційних даних



# 4 Прикладні аспекти захисту аутентифікаційних даних

З метою автоматизації частини процесів і спрощення роботи користувачів існує можливість використання плагінів (plugins).



# 4 Прикладні аспекти захисту аутентифікаційних даних

З метою автоматизації частини процесів і спрощення роботи користувачів існує можливість використання плагінів (plugins).

**DataBaseBackup** – допомагає в автоматичному режимі створювати резервну копію бази даних.
**FaviconDownloader** – цей плагін завантажує та зберігає favicons (позначки). Favicons це маленький значок/логотип, який використовується для ідентифікації багатьох веб-сайтів, які зазвичай відображаються в адресному рядку браузера, списку закладок і на вкладках.
**HIBPOfflineCheck** - автономні перевірки діючих паролів бази даних щодо файлу скомпрометованих паролів в HaveIBeenPwned passwords.
**WebAutoType** - цей плагін дозволяє виконувати автоматичний запуск на основі поточної URL-адреси веб-браузера замість заголовка вікна. Підтримуються різні браузери (Internet Explorer, Firefox, Opera, Chrome, ...).

# 4 Прикладні аспекти захисту аутентифікаційних даних

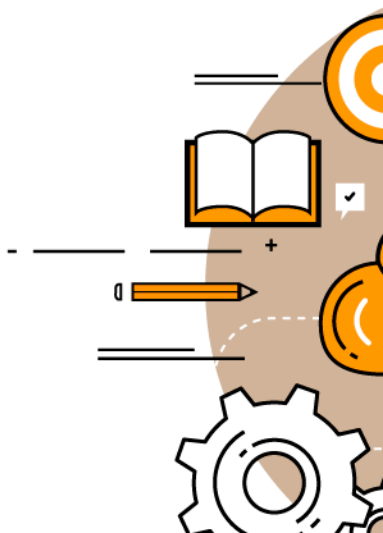
**WinKee** – плагін допомагає безпечно автоматизувати запуск ПЗ KeePass. WinKee окремо шифрує, зберігає і отримує облікові дані доступу до бази даних KeePass у файлі конфігурації KeePass (KeePass.config.xml). Пароль та ключовий шлях до файлу будуть зашифровані на основі облікових даних облікового запису Windows.

Оскільки при використанні плагіна WinKee ми зберігаємо облікові дані KeePass (зашифровані AES-256) в конфігураційному файлі, то для випадку його його компрометації можна використовувати вбудовану в операційну середу функцію шифрування за допомогою згенерованого сертифікату.

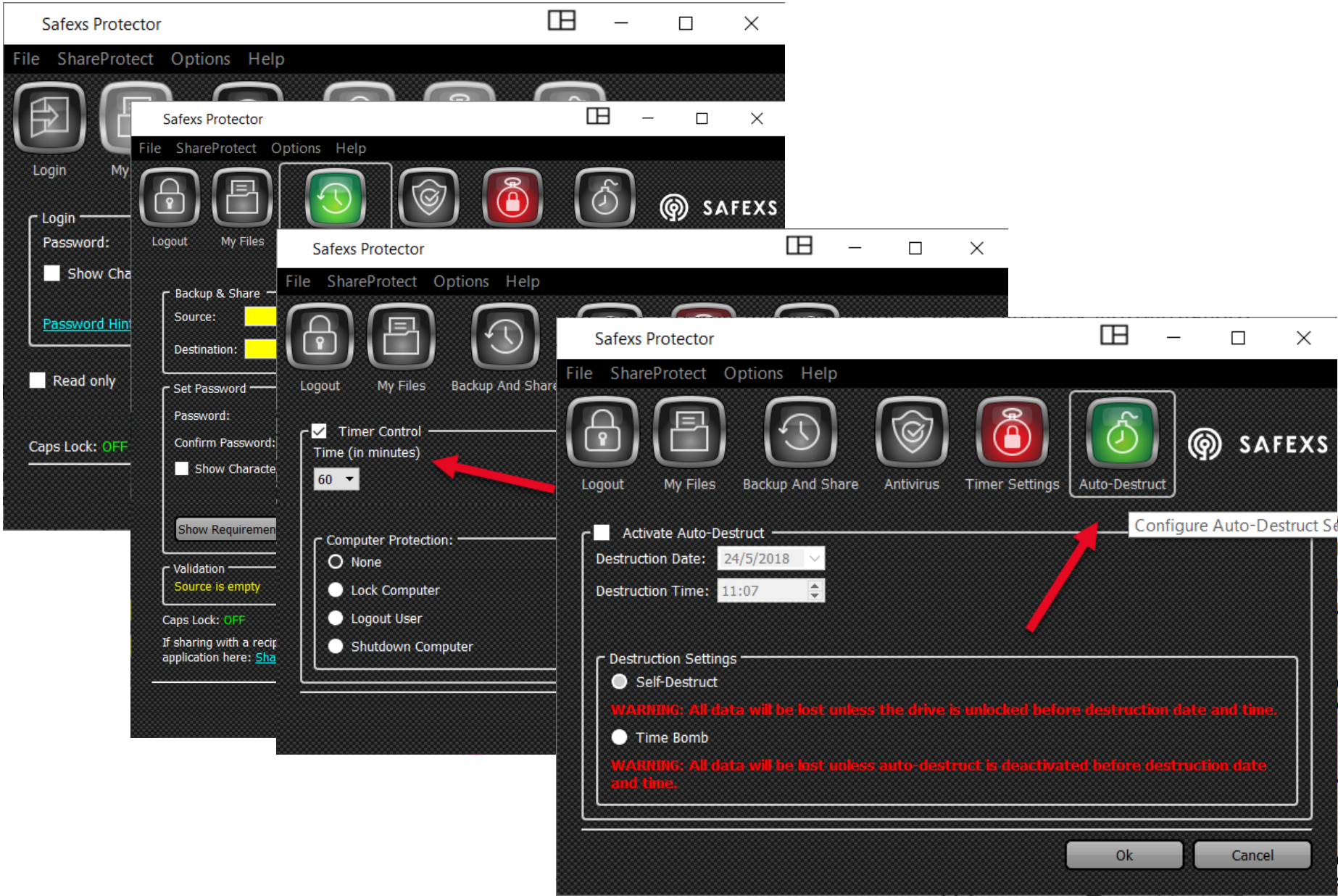
Сертифікат краще зберігати на захищеному електронному носії **eToken**. Це дає ще один рівень захисту аутентифікаційних даних.

# 4 Прикладні аспекти захисту аутентифікаційних даних


Safexs Protector XT - інструмент для захисту даних в компактному формфакторі USB-накопичувача, має великий функціонал безпеки для захисту від несанкціонованого доступу, вірусів і втрати даних.



# 4 Прикладні аспекти захисту аутентифікаційних даних



# 4 Прикладні аспекти захисту аутентифікаційних даних

Об'єднуючи два рішення - сервіс для зберігання паролів KeePass (Open Source) з програмним шифруванням і апаратний захист (захищена USB-флешка) ми забезпечуємо можливість безпечного зберігання аутентифікаційних даних користувачів.