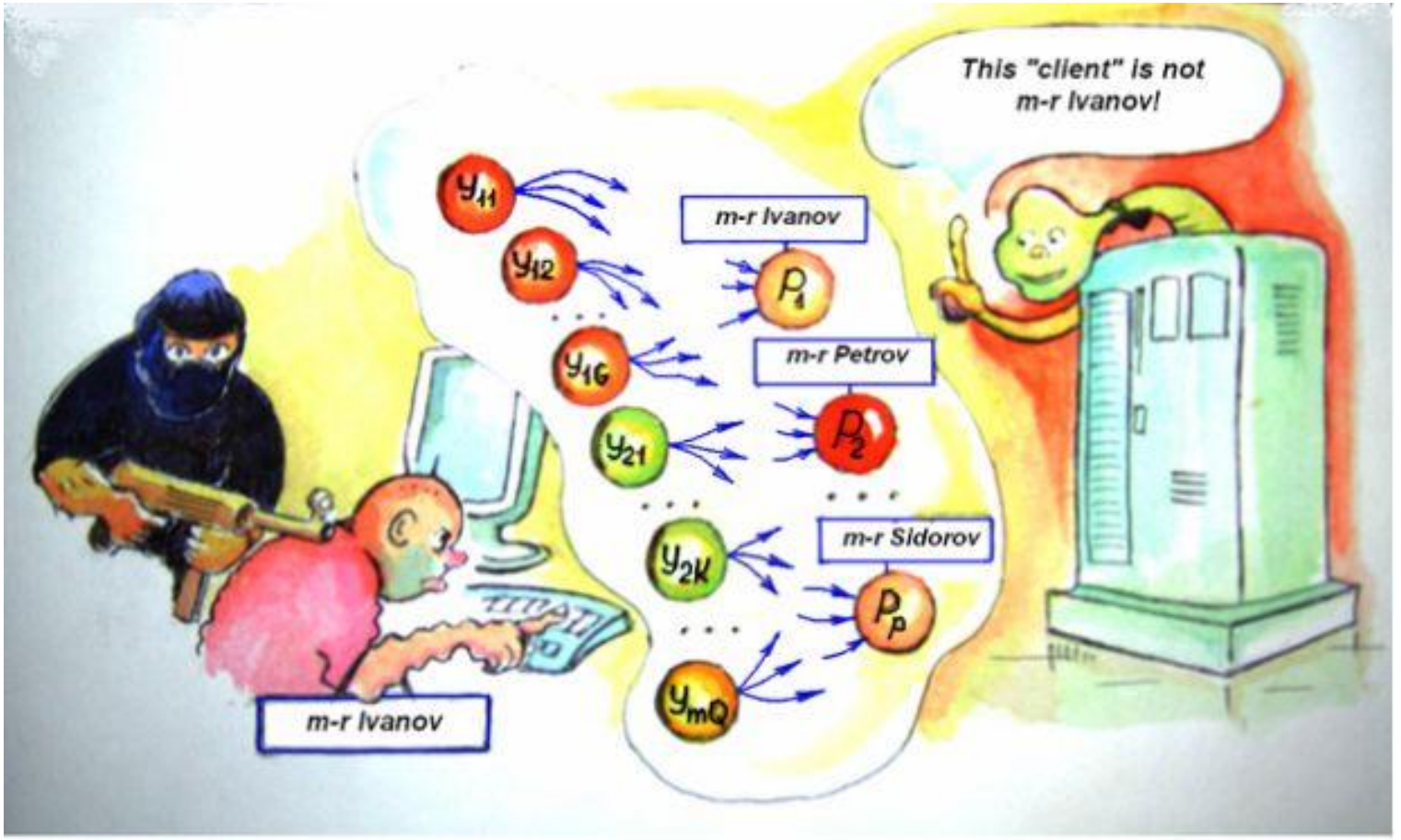
Захист інформації в
телекомунікаційних системах

ВЗАЄМНА АВТЕНТИФІКАЦІЇ СУБ’ЄКТІВ

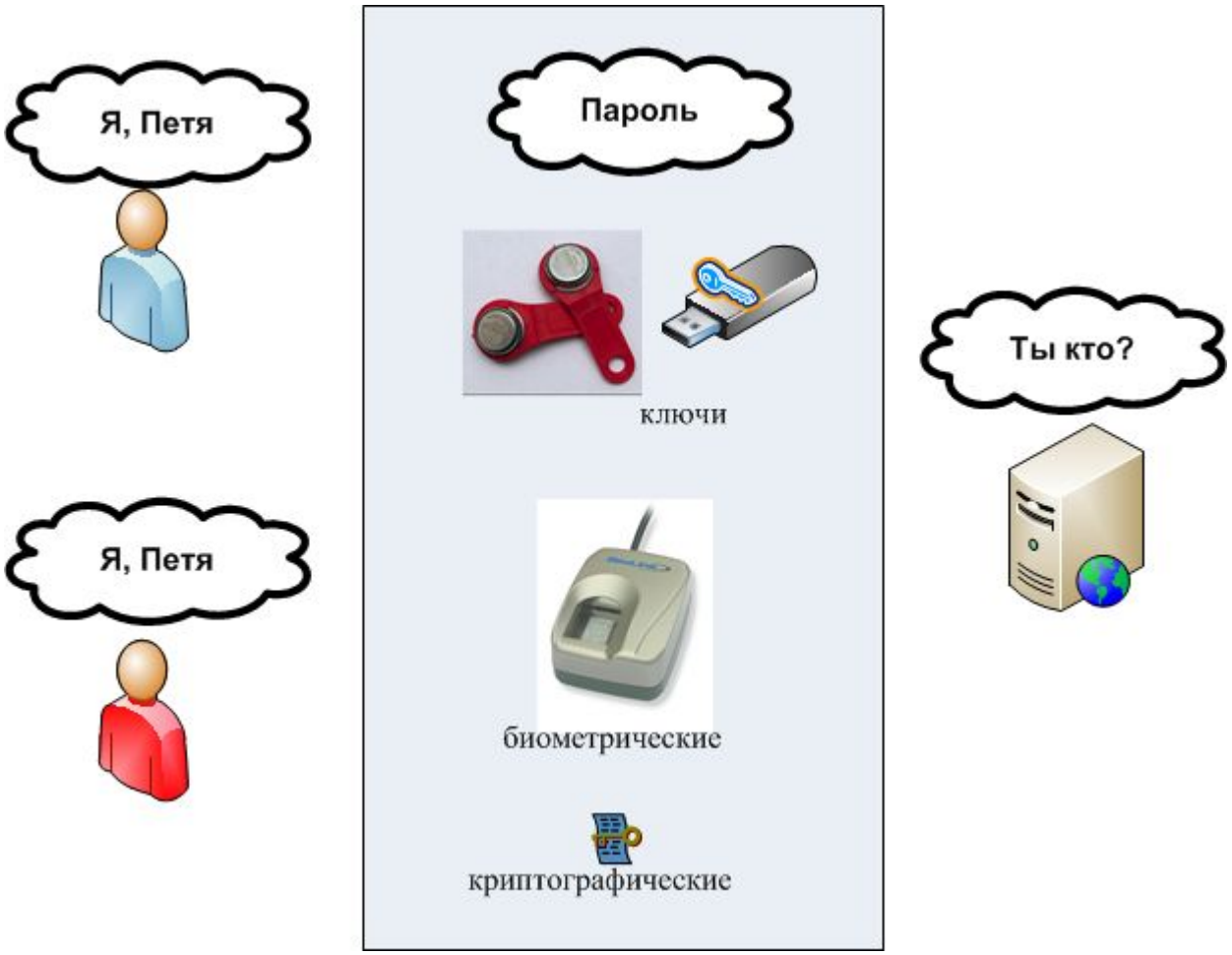
Лекція № 10

Розробив: доц., к.т.н. Золотарьов В.А.
Харківський національний університет радіоелектроніки
Факультет інфокомунікацій
Кафедра ІМІ

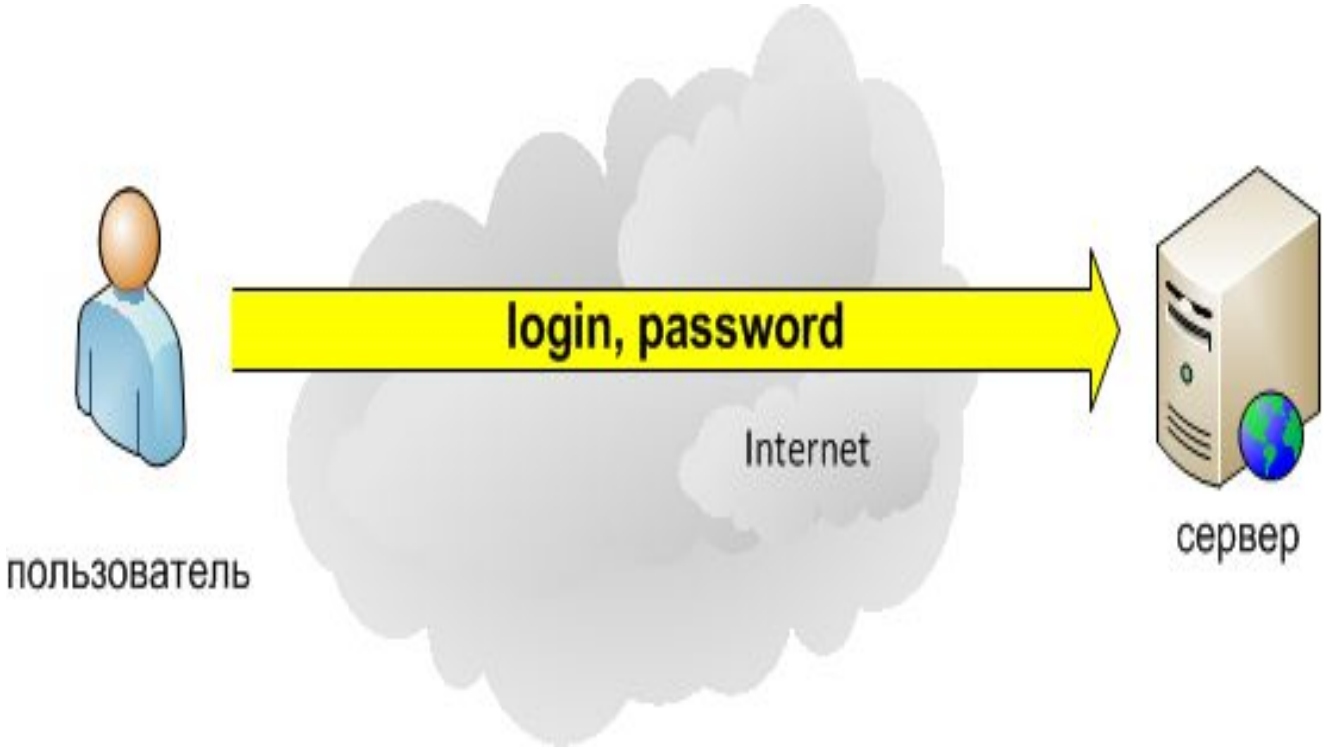
1. Парольні системи



Методи автентифікації



PAP (Password Authentication Protocol)



Парольна система

це програмно-апаратний комплекс, що реалізує системи ідентифікації та автентифікації користувачів інформаційно-телекомунікаційних систем на основі одноразових чи багаторазових паролів.

Парольна система

- функціонує разом з підсистемами розмежування доступу і реєстрації подій.
- може виконувати ряд додаткових функцій, зокрема генерацію і розподіл сеансових криптографічних ключів.

Основні компоненти парольної системи:

- інтерфейс користувача;
- інтерфейс адміністратора;
- модуль сполучення з іншими підсистемами безпеки;
- база даних облікових записів.

Розкриття параметрів облікового запису через:

- підбір в інтерактивному режимі;
- підглядання;
- навмисну передачу пароля його власником іншій особі;
- захоплення бази даних парольної системи (якщо паролі не зберігаються в базі **у** відкритому вигляді, для їхнього відновлення може знадобитися підбір чи дешифрування);
- перехоплення переданої по мережі інформації про пароль;
- збереження пароля в доступному місці.

.

Втручання у функціонування компонентів парольної системи через:

- Впровадження програмних закладок
- Виявлення і використання помилок, допущених на стадії розробки;

Недбалість користувача, який може

- вибрати пароль, що легко запам'ятати і також легко підібрати;
- записати пароль, що складно запам'ятати, і покласти запис у доступному місці;
- увести пароль так, що його зможуть побачити сторонні;
- передати пароль іншій особі навмисно чи під впливом омани,

Вимоги щодо вибору пароля	Отриманий ефект
Встановлення мінімальної довжини пароля	Ускладнює завдання зловмисника при спробі підглянути або підібрати пароль методом «тотального випробування»
Використання в паролі різних груп символів	Ускладнює завдання зловмисника при спробі підібрати пароль методом «тотального випробування»
Перевірка і відбраковування пароля за словником	Ускладнює завдання зловмисника при спробі підібрати пароль за словником

Вимоги щодо вибору пароля	Отриманий ефект
Встановлення максимального терміну дії пароля	Ускладнює завдання зловмисника при спробі підібрати пароль методом «тотального випробування», в тому числі без безпосереднього звернення до системи захисту (режим off-line)
Встановлення мінімального терміну дії пароля	Перешкоджає спробам користувача замінити пароль на старий після його зміни за попередньою вимогою
Ведення журналу історії пароля	Забезпечує додатковий ступінь захисту за попередньою вимогою

Вимоги щодо вибору пароля	Отриманий ефект
Застосування евристичного алгоритму, що відкидає паролі за даними журналу історії	Ускладнює завдання зловмисника при спробі підібрати пароль за словником або з використанням евристичного алгоритму
Обмеження кількості спроб введення пароля	Перешкоджає інтерактивному підбору паролів зловмисником
Підтримка режиму примусової зміни пароля користувача	Перешкоджає інтерактивному підбору паролів зловмисником

Вимоги щодо вибору пароля	Отриманий ефект
Використання затримки при введенні неправильного пароля	Перешкоджає інтерактивному підбору паролів зловмисником
Заборона на вибір пароля самим користувачем і автоматична генерація паролів	Виключає можливість підібрати пароль за словником. Якщо алгоритм генерації паролів невідомий зловмиснику, останній може підібрати пароль тільки методом «тотального випробування»
Примусова зміна пароля при першій реєстрації користувача в системі	Захищає від неправомірних дій системного адміністратора, який має доступ до пароля в момент створення облікового запису

Недоліки парольних систем

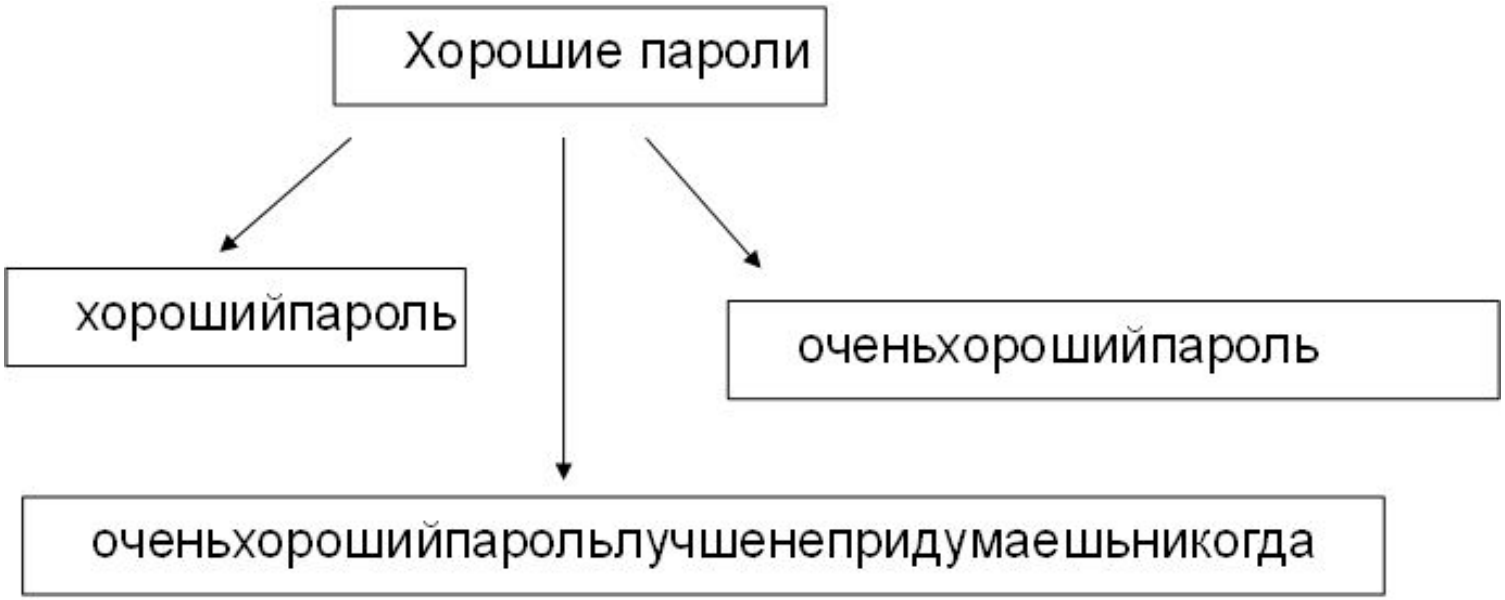
- пароль необхідно періодично змінювати;
- він не повинен легко асоціюватися з користувачем; пароль необхідно запам'ятовувати (але тоді його можна забути) чи записувати (але тоді запис може потрапити до зловмисника);
- якщо пароль розкритий, то не існує ефективного способу виявлення, ким він далі використовувався - користувачем чи зловмисником;
- парольний файл повинен ретельно захищатися.

Погані паролі

Чому ці паролі погані?

Пароль	Слабкість
2	Невелика кількість символів – легко перебрати
123456	Один з популярних паролів (номер телефону, дата народження і т.п.)
Пароль	Словарне слово - перебирають шляхом перебору
"Gjhs6129dgGF_9eK_sj 2vc9d"	Пароль надзвичайно складний – неможливо запам`ятати

Добрі паролі



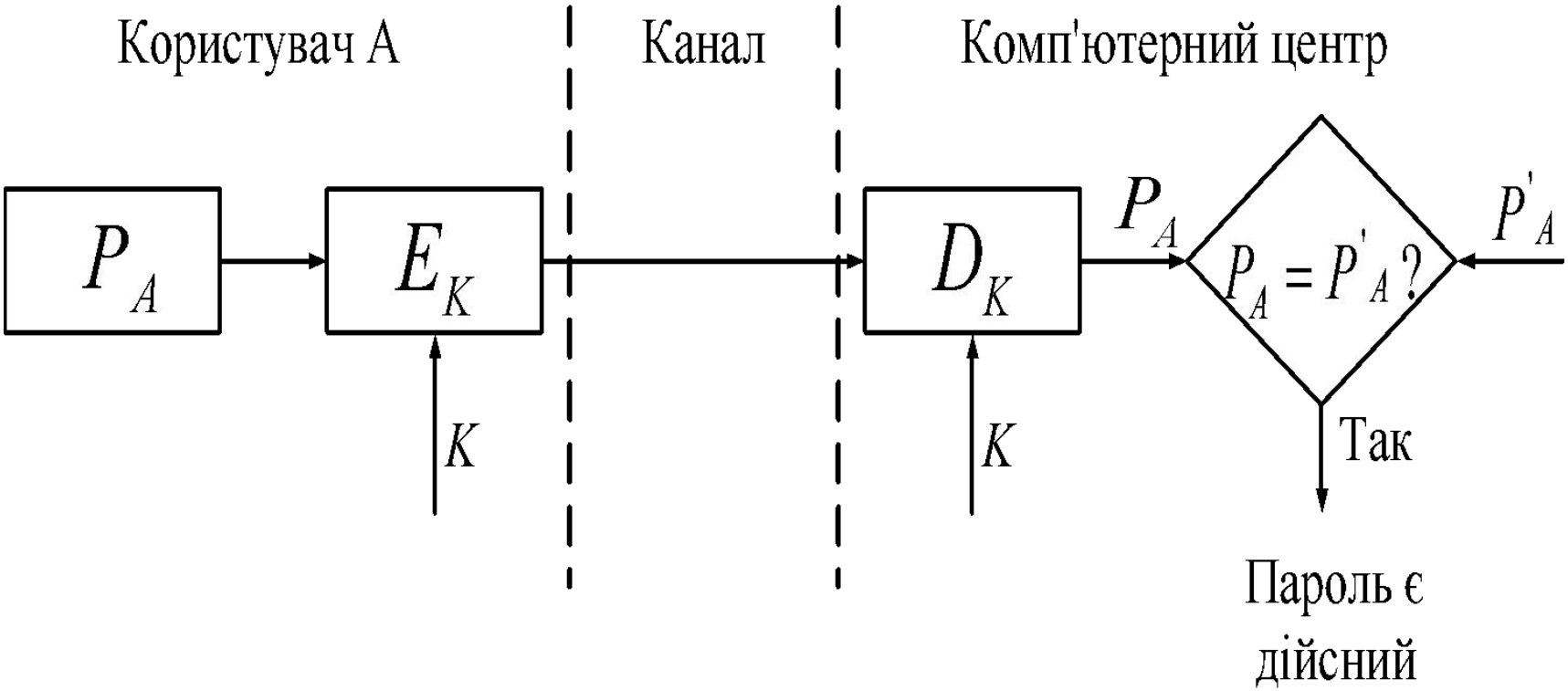
Найкращі паролі побудовані на фразах

- Легко запам`ятовуються
- Достатньо довгі
- Не піддаються словарній атаці

Розв`язання проблеми «перегляд паролів у системі»

- Шифрування
- На диску зберігається не сам пароль, а його контрольна сума або хеш-функція

Схема простої автентифікації за допомогою пароля



Односпрямована функція паролю

$$\alpha(P) = E_P(ID)$$

де P – пароль відправника;
ID – ідентифікатор відправника;
Ep - процедура шифрування, виконувана з використанням пароля P як ключ.

Інший спосіб завдання односпрямованої функції

$$\alpha(P) = E_{P \oplus K}(ID)$$

де K і ID - відповідно ключ і ідентифікатор відправника.

Схема автентифікації за допомогою пароля з використанням ідентифікаційної таблиці

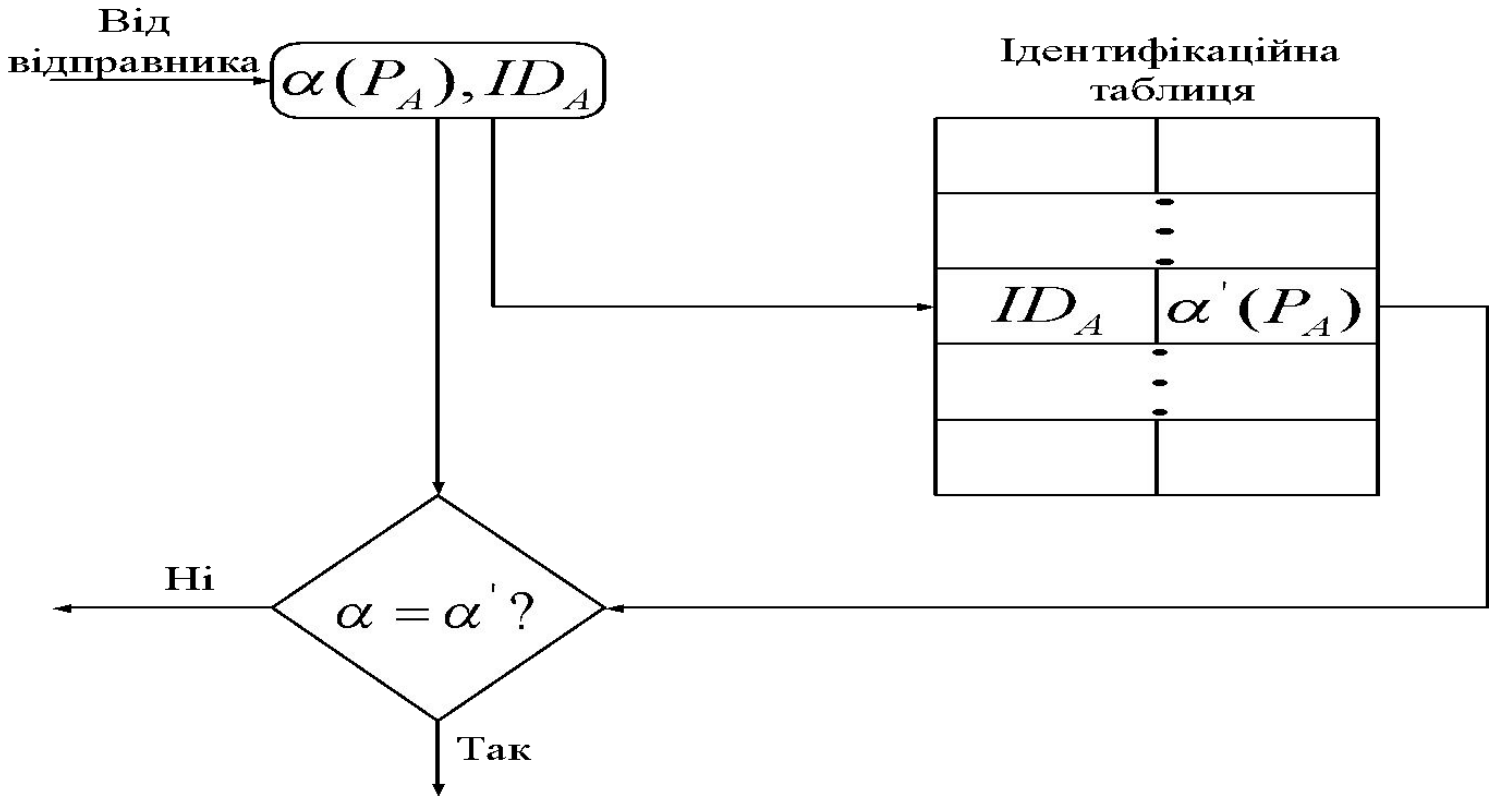
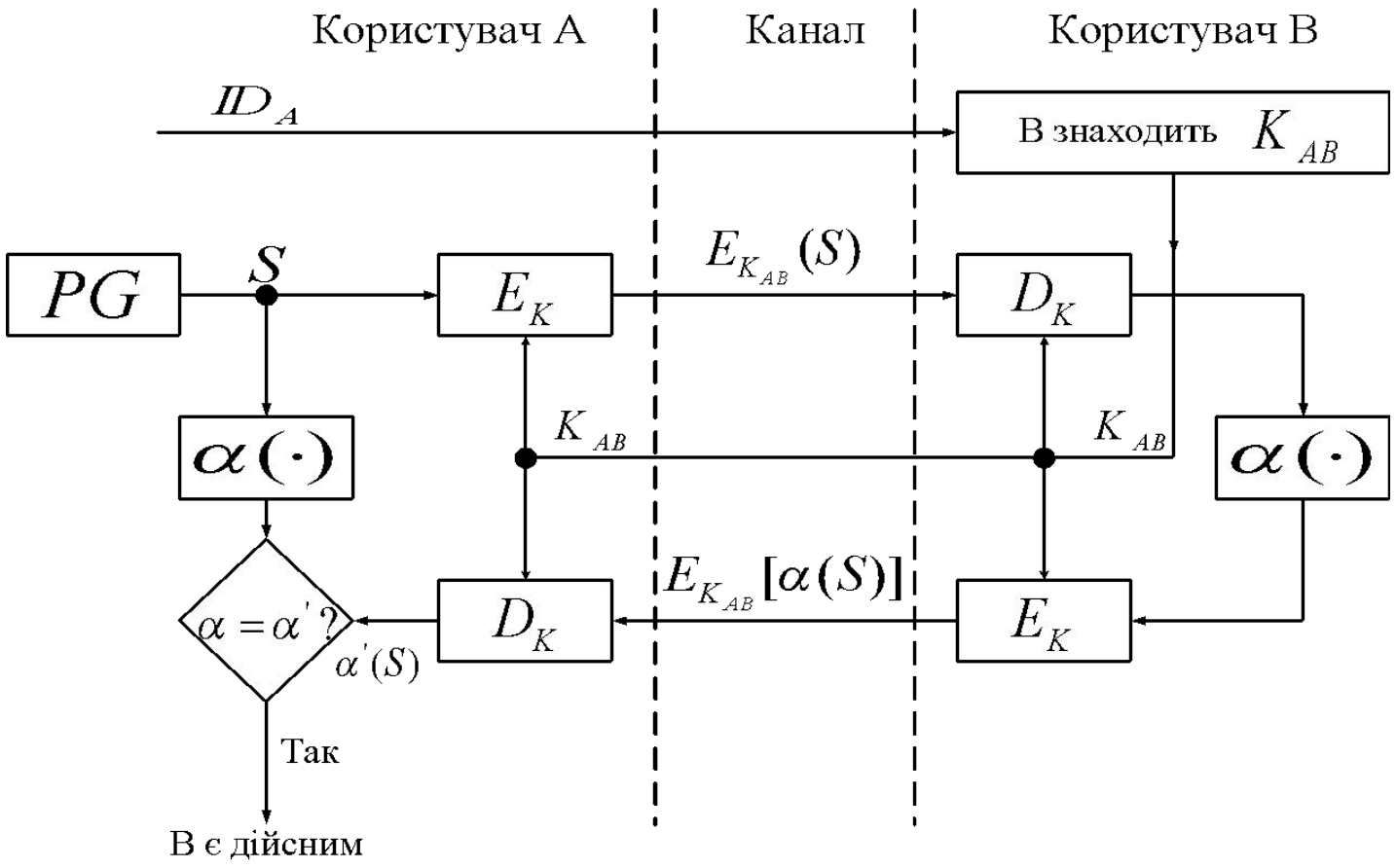


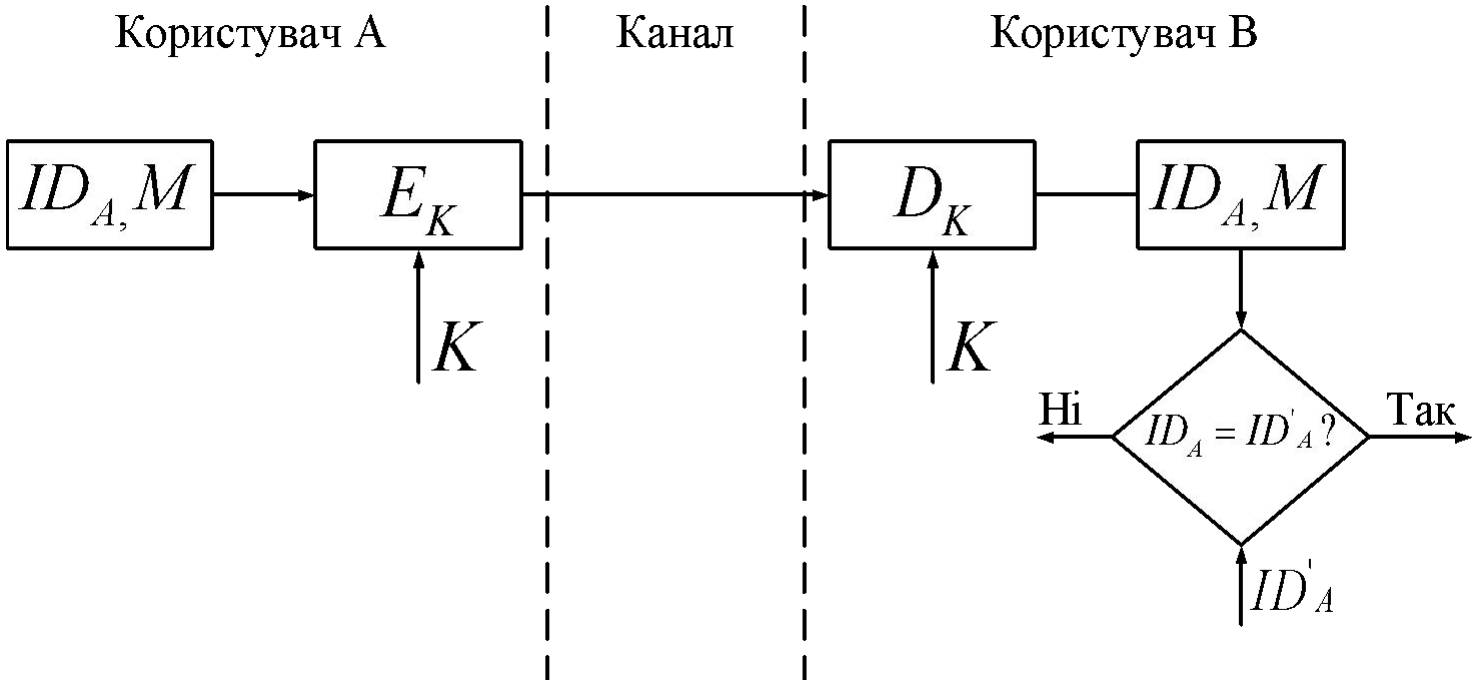
Схема процедури рукостискання (користувач А перевіряє справжність користувача В).



Передана криптограма має вигляд:

$$E_K(ID_A, M)$$

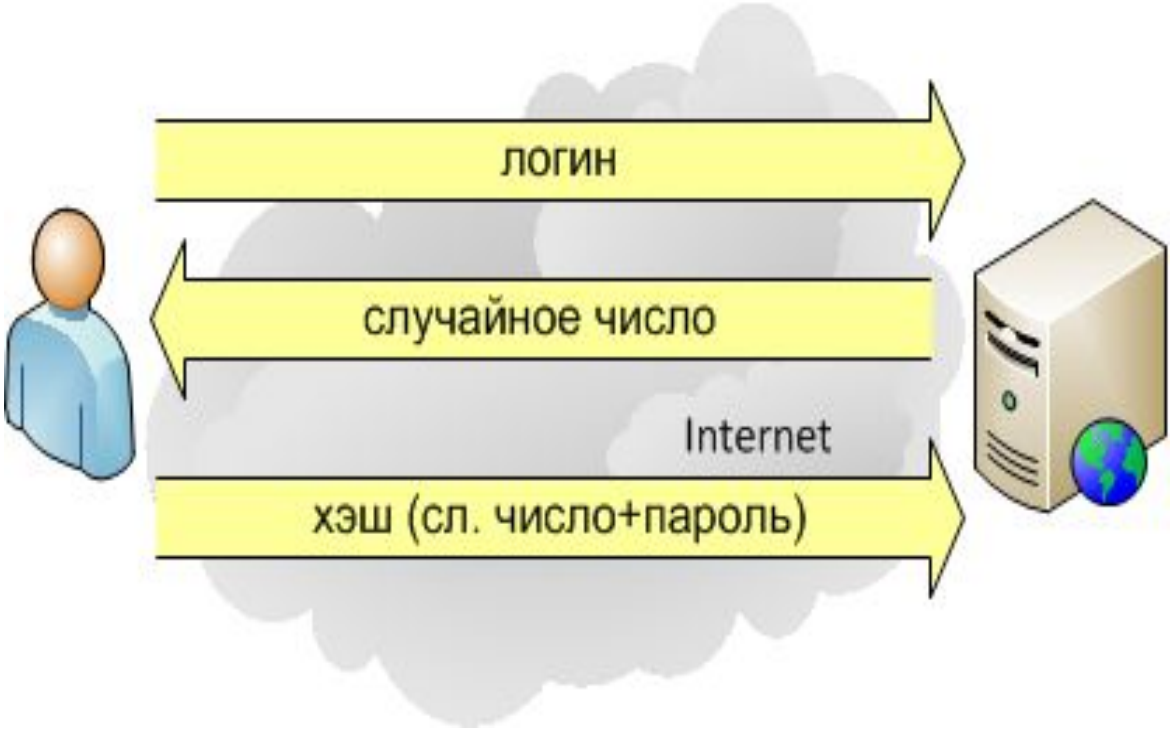
Схема безперервної перевірки автентичності відправника



Питання 2

Протоколи автентифікації «запит-відповідь»

CHAP (Challenge Handshake Authentication Protocol)

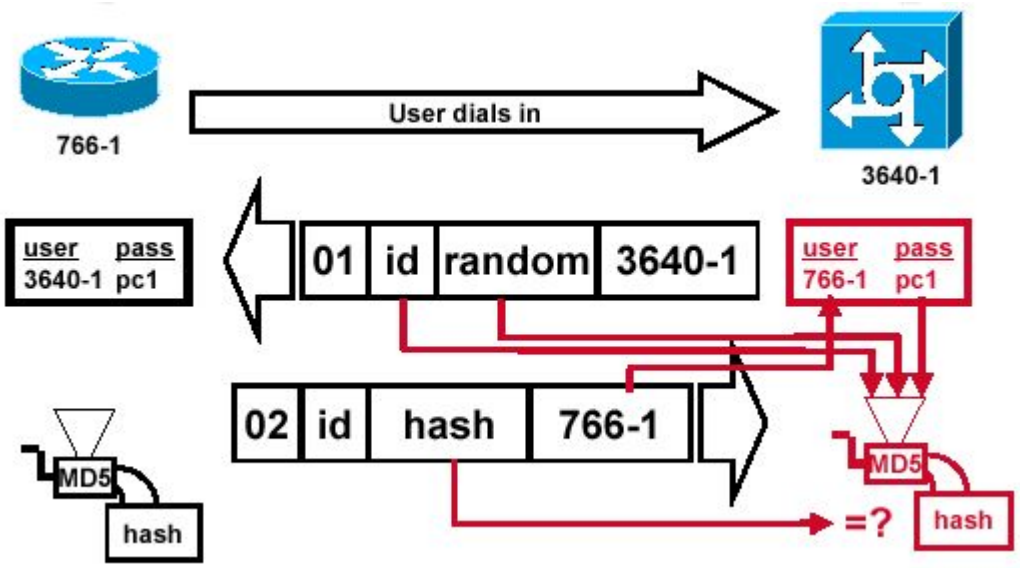


Алгоритм CHAP (автентифікація без передавання паролю)

1. Користувач надсилає серверу запит на доступ (login)
2. сервер відправляє клієнту випадкове число
3. За допомогою цього випадкового числа та пароля користувач обчислює г’еш
4. Клієнт надсилає г’еш серверу
5. Сервер порівнює надісланий г’еш зі своїм обчисленням
6. У випадкові проміжки часу сервер відправляє новий запит та повторює кроки з 2 по 5.

Недолік CHAP

- Необхідність зберігання паролю на сервері



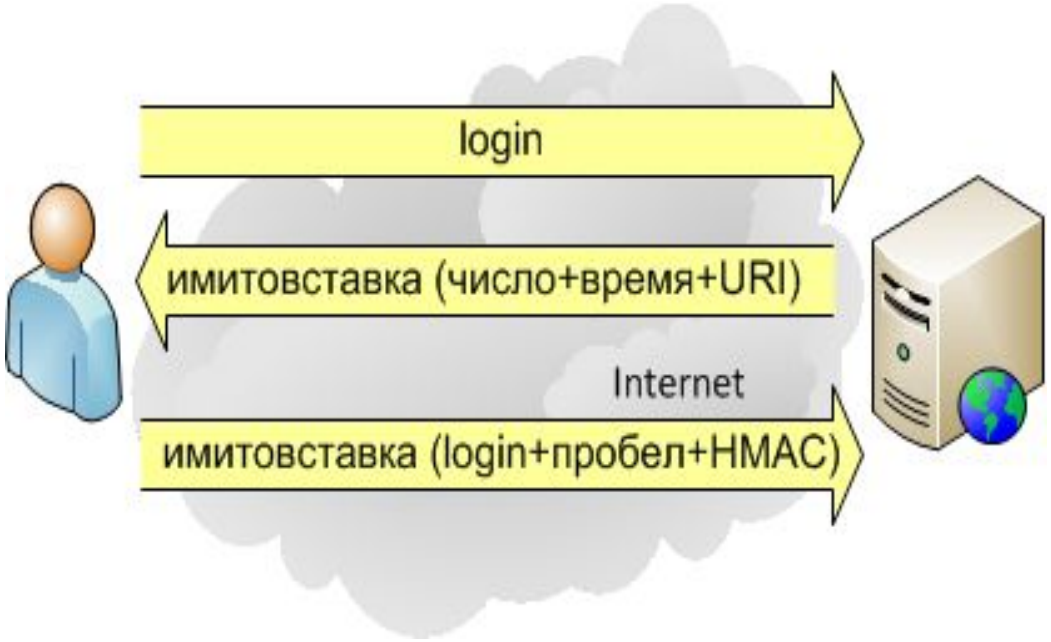
CRAM - (challenge-response authentication mechanism)

- Ґрунтується на обчисленні імітовставки за алгоритмом HMAC, роль симетричного ключа виконує пароль.
- У залежності від алгоритму Ґешування - CRAM-MD5, CRAM-MD4, CRAM-SHA1 і т.п.

Алгоритм CRAM

1. Користувач надсилає серверу запит на доступ (login)
2. Сервер обчислює імітовставку з таємним ключем – паролем користувача для рядка (випадкове число + часова мітка + доменне ім`я сервера) (наприклад: <1896.697170952@postoffice.reston.mci.net>)
3. Сервер надсилає клієнтові імитовставку
4. Клієнт обчислює імітовставку з рядка
- (ідентифікатор клієнта (login) + пробіл + імітовставка сервера)
5. Відправляє серверу
6. Сервер перевіряє отримане повідомлення з очикуваним

Протокол CRAM (пароль на сервері може зберігатися в геш)



Digest access authentication (DIGEST-MD5)

1. Запит клієнта (без автентифікації)
2. Відповідь сервера (Unauthorized), що містить
"realm" - рядок (наприклад : realm=testrealm@host.com)
"nonce" – випадкове число сервера (наприклад: nonce="dcd98b7102dd2f0e8b11d0f600bfb0c093")
3. Клієнт обчислює г'еш HA1 = MD5 (username: realm: password)
4. Клієнт обчислює г'еш HA2 = MD5 (URI)
5. Клієнт обчислює г'еш для відповіді Response = MD5(HA1:nonce:nc:cnonce:qop:HA2)
"nc" - значення рахівника запитів
"cnonce" - клієнтське випадкове значення
"qop" - код якості відповіді
6. Клієнт надсилає відповідь
7. Сервер порівнює отримане та обчислене значення

Взаємна автентифікація

- Клієнт відправляє запит серверу, що містить його login і випадкове число N1
- Сервер зашифровує число N1, генерирує випадкове число N2, і відправляє оба їх клієнту
- Клієнт розшифровує числа (N1,N2) та порівнює перше (N1) число з N1. Ідентичність означає, що сервер має той самий ключ що і клієнт
- Клієнт зашифровує число N2 і результат відправляє серверу
- Сервер розшифровує повідомлення. При співпадінні результату з вихідним числом N2, взаємна автентифікація відбулася.

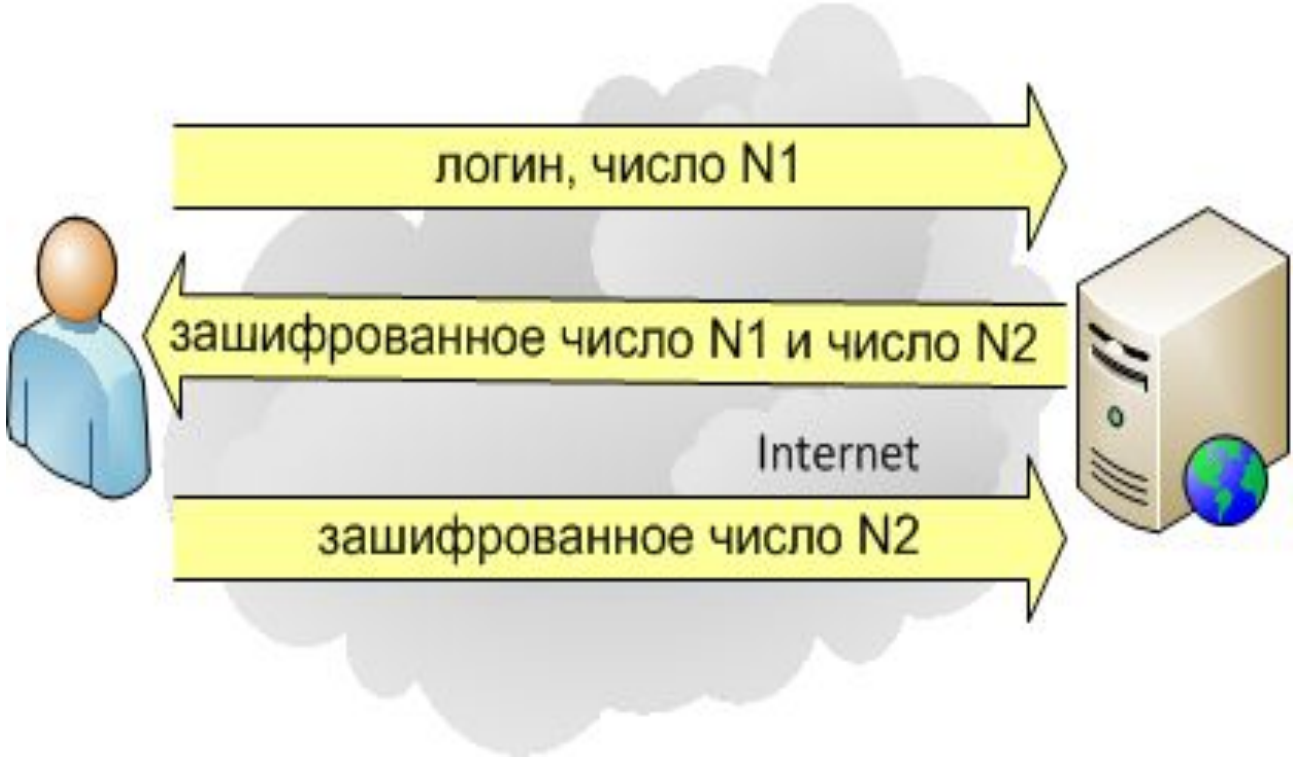
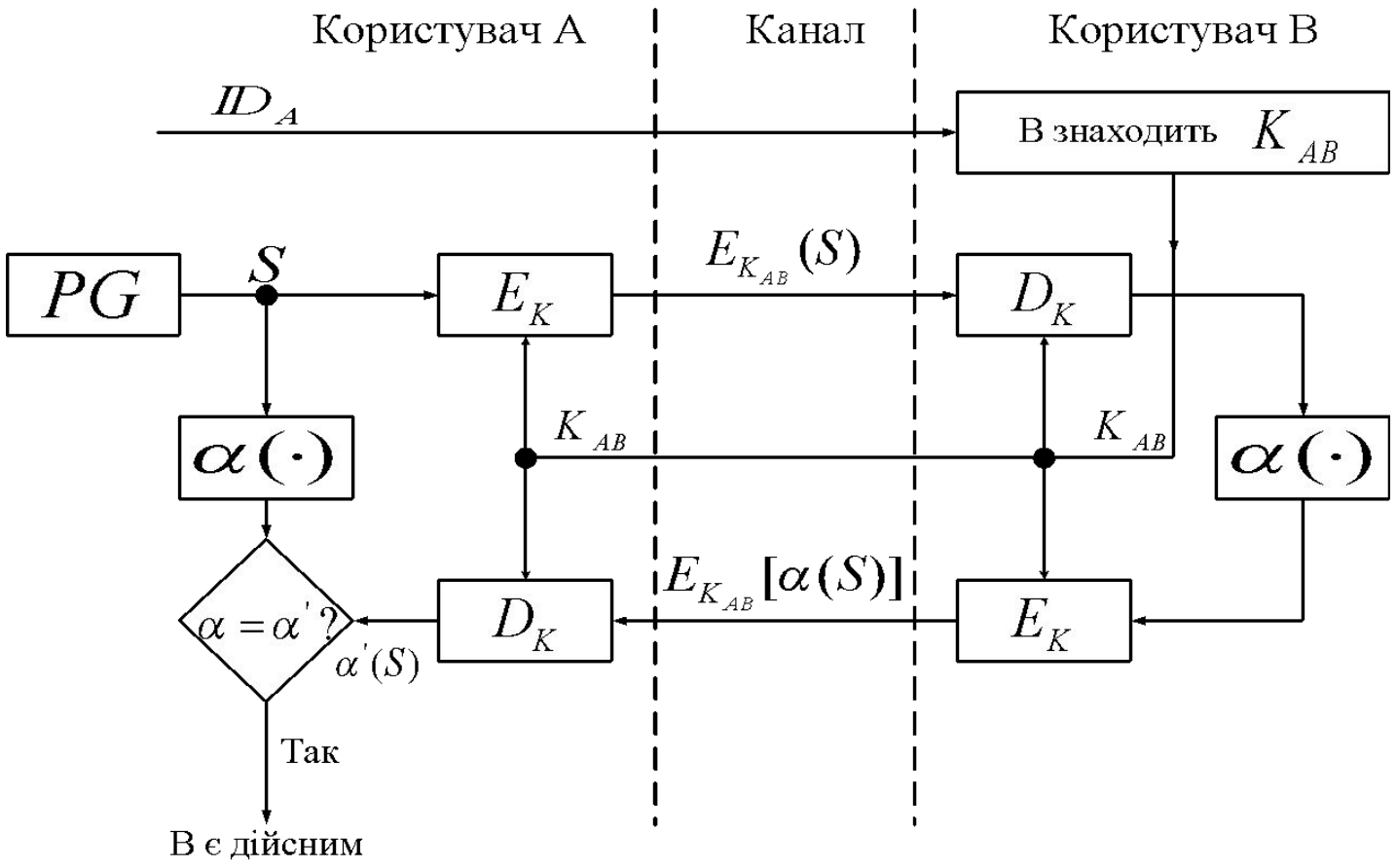

логин, число N1
зашифрованное число N1 и число N2
Internet
зашифрованное число N2

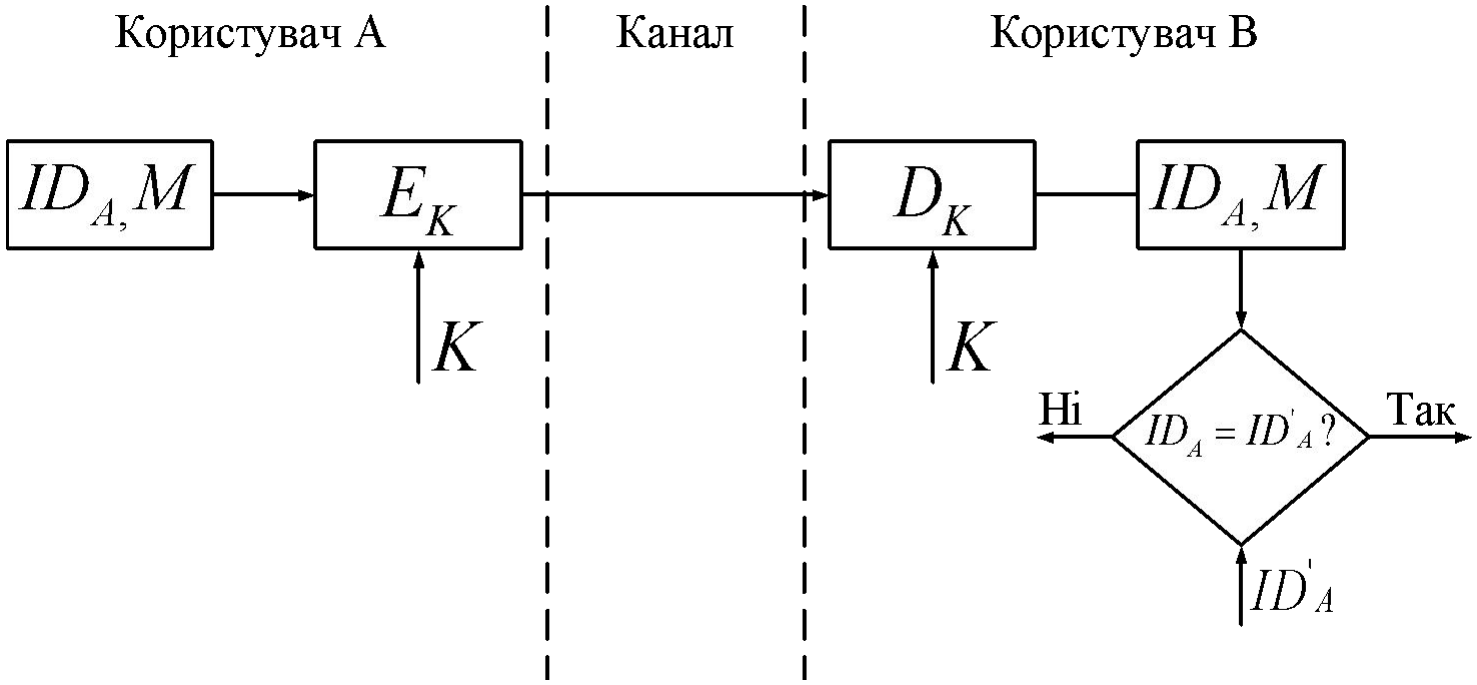
Схема процедури рукостискання (користувач А перевіряє справжність користувача В).



Передана криптограма має вигляд:

$$E_K(ID_A, M)$$

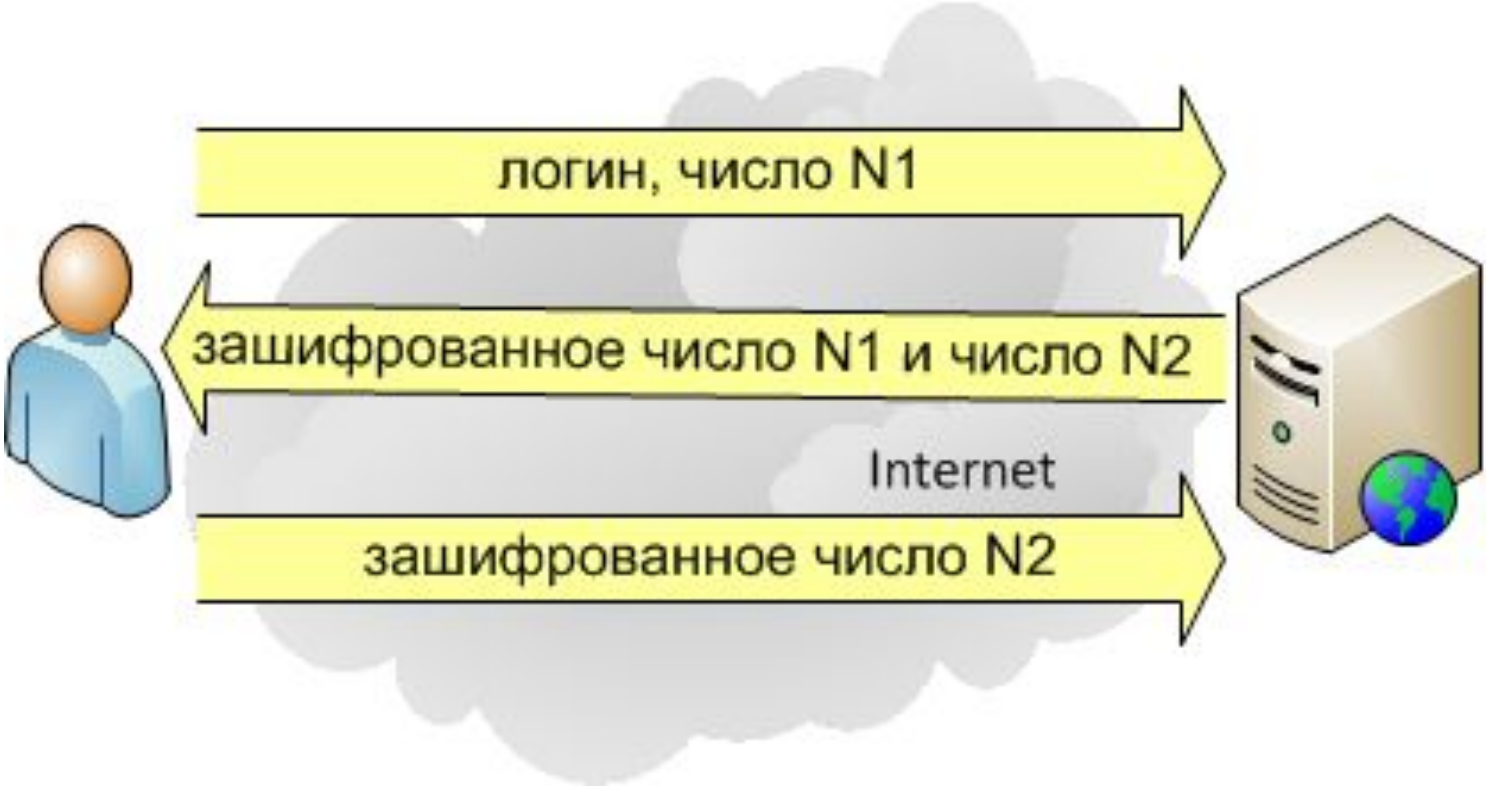
Схема безперервної перевірки автентичності відправника



Взаємна автентифікація

- Клієнт відправляє запит серверу, що містить його login, випадкове число N1
- Сервер зашифровує число N1, генерує випадкове число N2, і відправляє їх обох клієнтові
- Клієнт розшифровує числа (N1,N2) та порівнює перше (N1) число з N1. Ідентичність означає, що сервер має той самий ключ що і клієнт
- Клієнт зашифровує число N2 і результат відправляє серверу
- Сервер розшифровує повідомлення. При співпадіні результату з вихідним числом N2, взаємна автентифікація відбулася.

Протокол взаємної автентифікації сервера



Автентифікація за одноразовими паролями (One-time password)

Існують різні підходи до створення одноразових паролів:

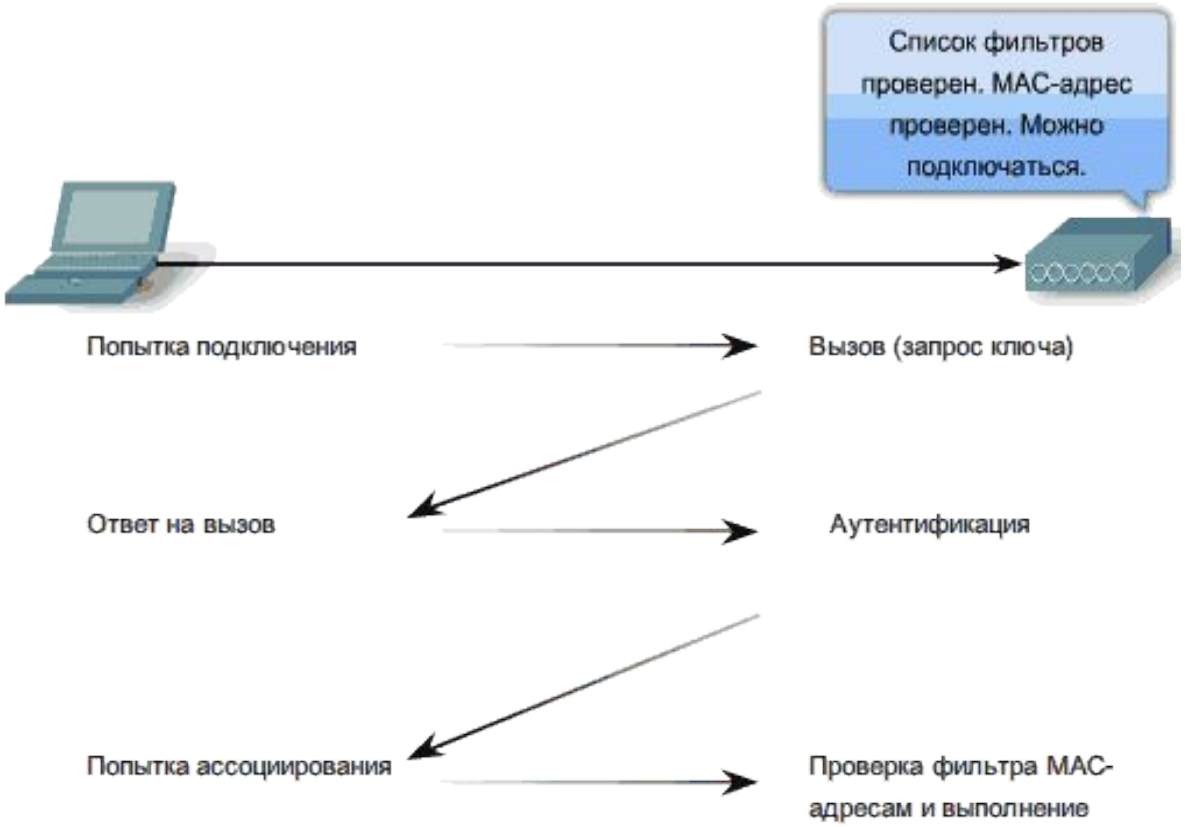
- Такі, що використовують математичні алгоритми задля створення нового пароля, ґрунтуючись на попередніх. Паролі фактично складають ланцюжок і мають бути використані у визначеному порядку.
- Такі, що ґрунтуються на часовій синхронізації між сервером і клієнтом, що забезпечує пароль. Пароль дійсний протягом короткого періоду часу
- Такі, що використовують математичний алгоритм в якому пароль заснований на запиті (наприклад, випадкове число, що обирається сервером, або частини вхідного повідомлення) та / або лічильника.

Одноразові паролі клієнти можуть отримати:

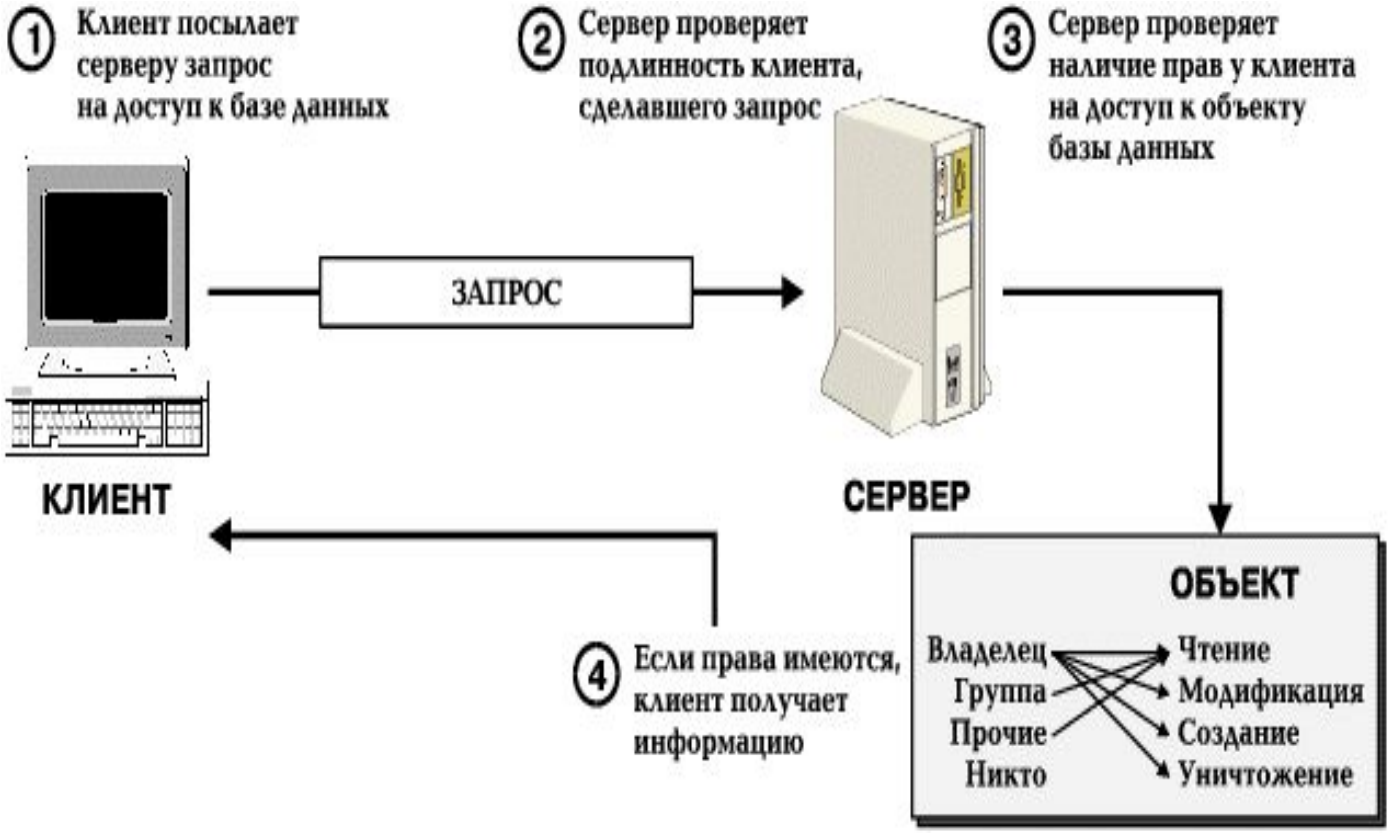
- На папері
- В токені
- Пересиланням (SMS)

Питання 2. Методи автентифікації

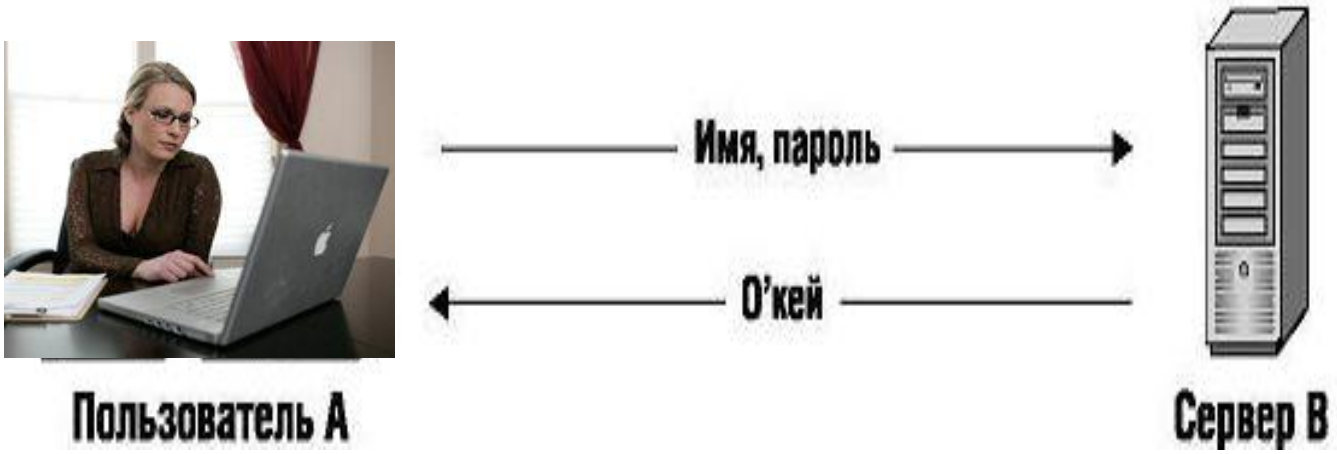
Відкрита автентифікація



Автентифікація в мережі Wlan


1 Клиент посылает серверу запрос на доступ к базе данных
2 Сервер проверяет подлинность клиента, сделавшего запрос
3 Сервер проверяет наличие прав у клиента на доступ к объекту базы данных
ЗАПРОС
КЛИЕНТ
СЕРВЕР
ОБЪЕКТ
Владелец
Группа
Прочие
Никто
Чтение
Модификация
Создание
Уничтожение
4 Если права имеются, клиент получает информацию

Автентифікація за допомогою протоколу



Автентифікація «запит – відповідь»

Автентифікація за допомогою неявного питання

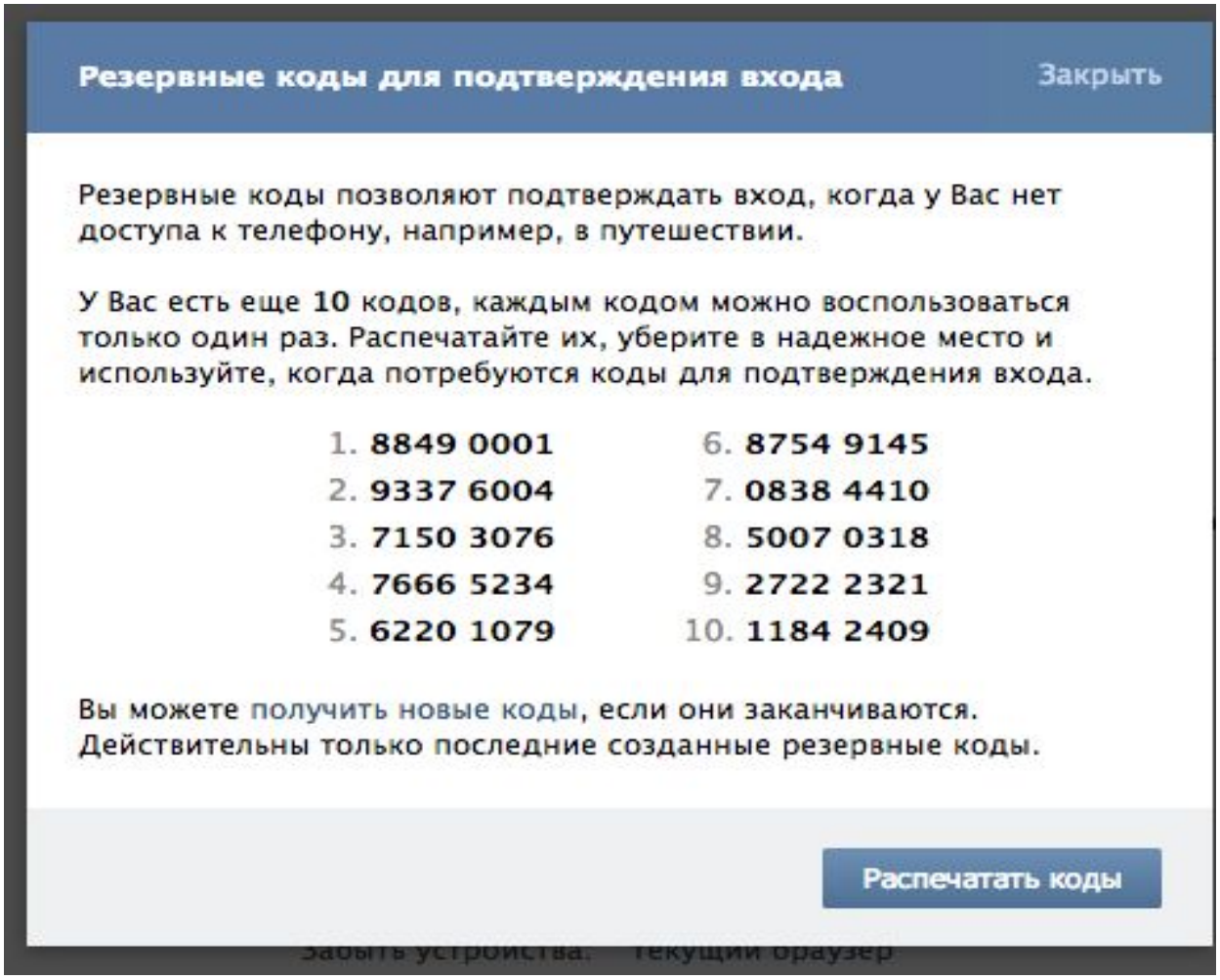
Двохфакторна авторизація

1-й етап – введення

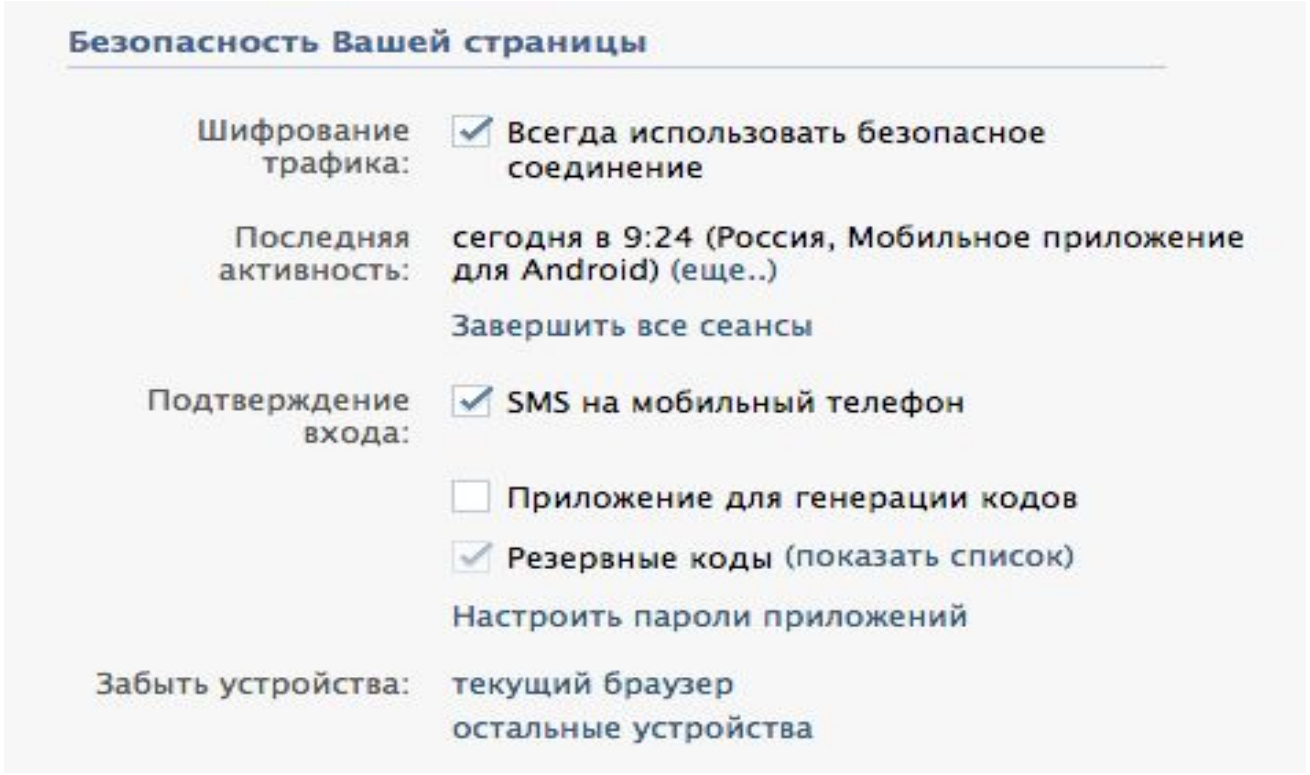
- Логин
- Пароль

2-й етап – користувач обирає один з 3-х варіантів

- Унікальний код по SMS
- Список резервних кодів, кожний з яких діє лише один раз
- Використовує спеціальні мобільні додатки для генерації кодів. Наприклад, Google Authentificator.


Резервные коды для подтверждения входа
Закрыть
Резервные коды позволяют подтверждать вход, когда у Вас нет доступа к телефону, например, в путешествии.
У Вас есть еще 10 кодов, каждым кодом можно воспользоваться только один раз. Распечатайте их, уберите в надежное место и используйте, когда потребуются коды для подтверждения входа.
1. 8849 0001
2. 9337 6004
3. 7150 3076
4. 7666 5234
5. 6220 1079
6. 8754 9145
7. 0838 4410
8. 5007 0318
9. 2722 2321
10. 1184 2409
Вы можете получить новые коды, если они заканчиваются.
Действительны только последние созданные резервные коды.
Распечатать коды

Для налаштування слід просканувати QR-код і ввести спеціальний код підтвердження



Перевірка справжності передбачає:

- використання неповторюваних блоків даних, в якості яких використовуються тимчасові мітки,
- механізми запит – відповідь;
- процедури рукостискання (handshake).

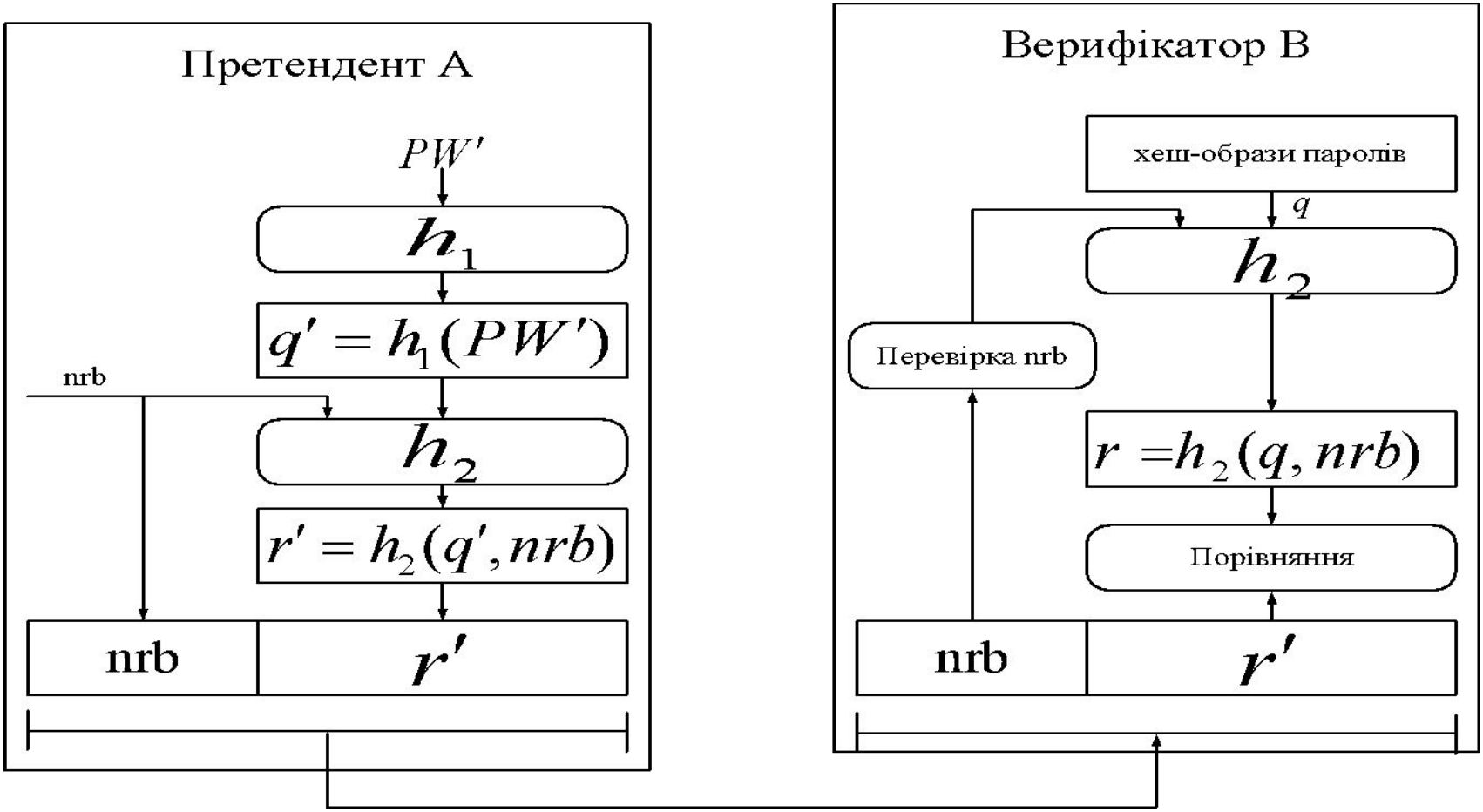
Використання позначок часу дозволяє

- реєструвати час відправлення конкретного повідомлення, що дає можливість одержувачеві визначити, наскільки «застаріло» отримане повідомлення, тобто захиститися від повтору.

Проблема допустимого часу затримки пов'язана

- з неможливістю миттєвої передачі повідомлення;
- з неможливістю абсолютної синхронізації ходу годинника одержувача і відправника.

Парольна схема автентифікації, захищена від повтору



Питання 3

Процедура рукостискання двох користувачів

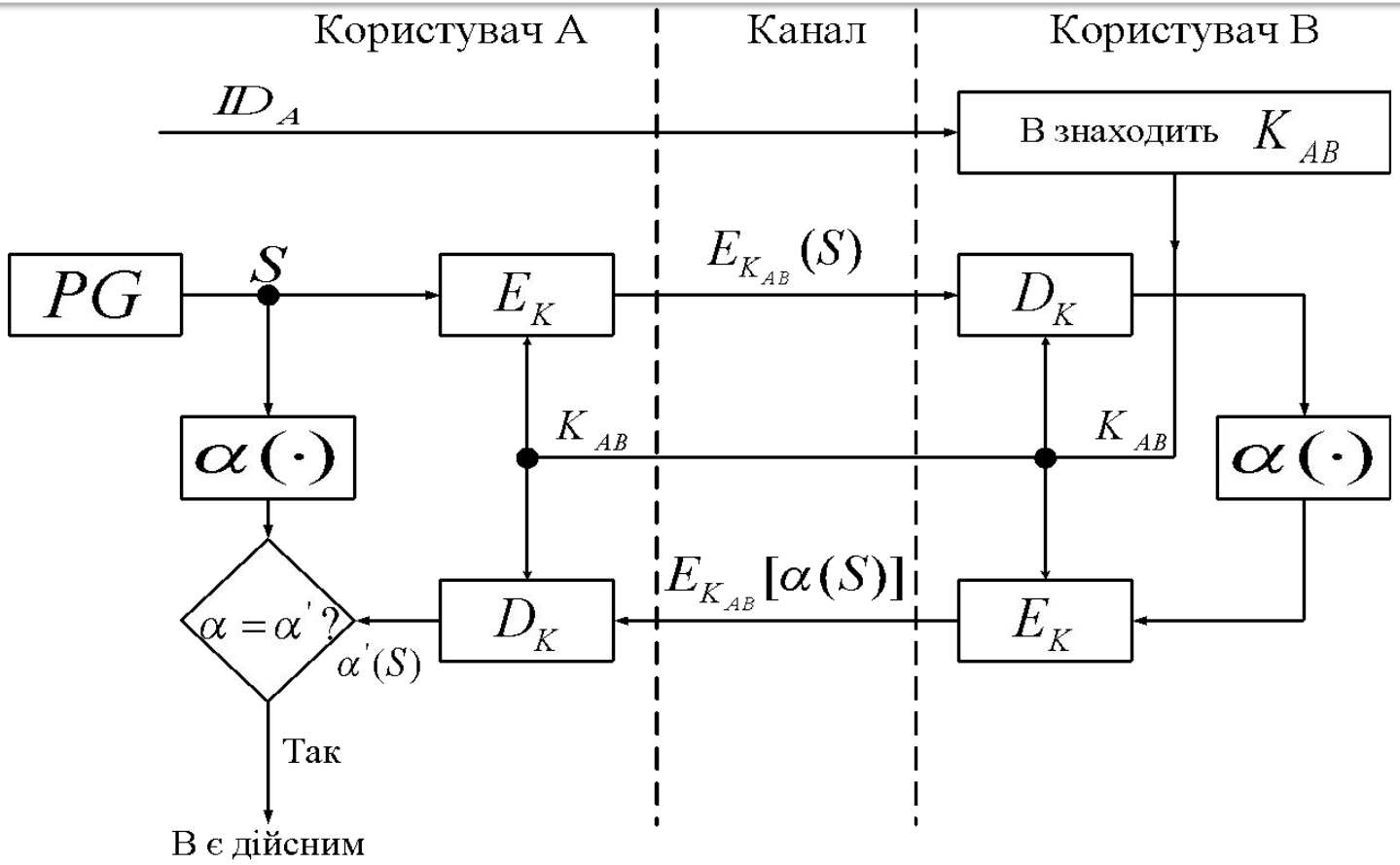
протокол Нідхема-Шрьодера

Застосовується симетрична криптосистема. Користувачі використовують однаковий таємний ключ Кав

Користувач Alice

Користувач Bob

Схема процедури рукостискання.



1-й крок: Alice

Alice ініціює процедуру рукостискання, відправляючи Bob свій ідентифікатор ID_A у відкритій формі.

2-й крок: Bob

- отримавши ідентифікатор ID_A, знаходить в базі даних секретний ключ K_{AB}
- вводить його в свою криптосистему.

3-й крок: Alice

- генерує випадкову послідовність S за допомогою псевдовипадкового генератора PG
- і відправляє її користувачеві B у вигляді криптограми

$$E_{Kab}(S)$$

4-й крок: Bob

- розшифровує цю криптограму і розкриває початковий вигляд послідовності S.

Крок 5: обидва користувачі перетворюють послідовність

s, використовуючи відкриту односпрямовану функцію α

$(.)$

Користувач Alice

Користувач Bob

6-й крок: Bob

шифрує повідомлення $\alpha(S)$ і відправляє цю криптограму Alice

7-й крок: Alice

- розшифровує цю криптограму
- порівнює отримане повідомлення $\alpha`(S)$ з вихідним $\alpha(S)$.
- Якщо ці повідомлення рівні, Alice визнає справжність Bob.

8-й крок: Bob

Перевіряє справжність Alice таким же чином

Безперервна перевірка автентичності відправника

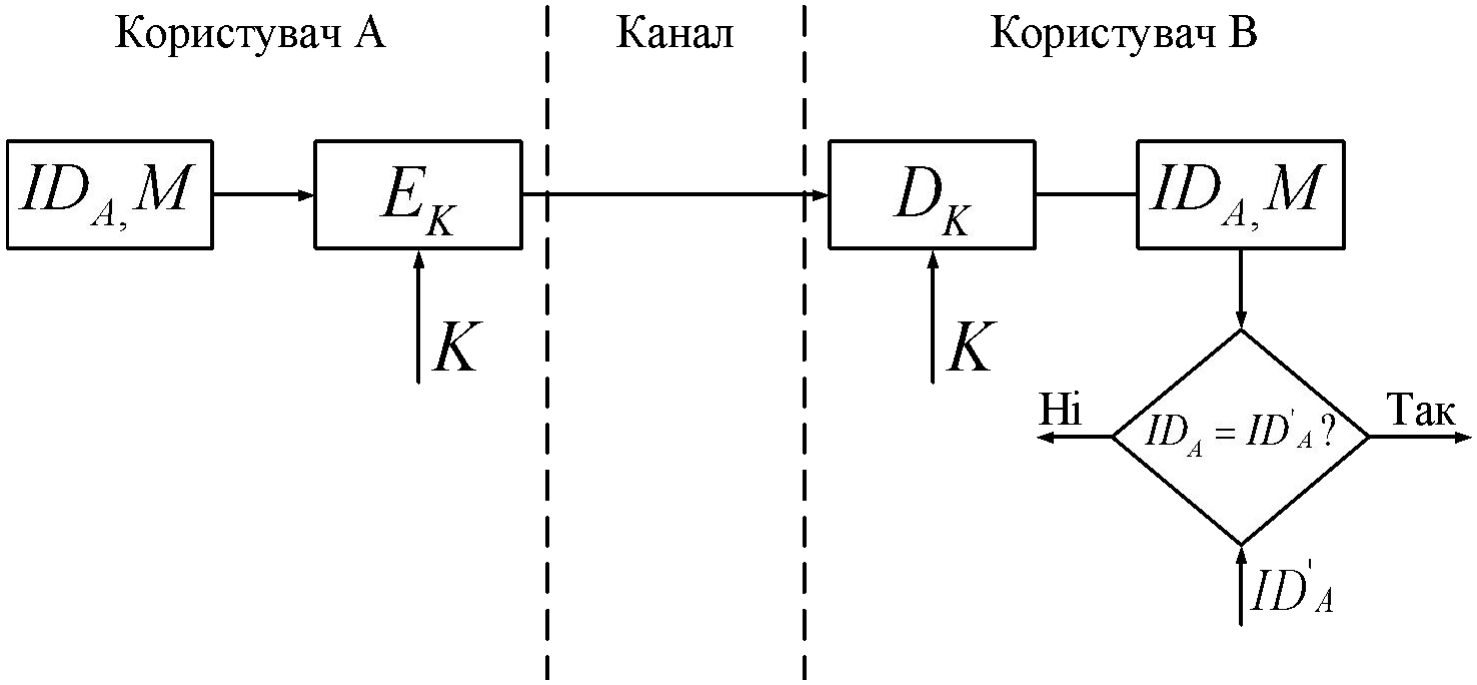
Користувач Alice

Користувач Bob

Передана криптограма має вигляд (ID_A – ідентифікатор відправника, M - повідомлення) :

$$E_K(ID_A, M)$$

Схема безперервної перевірки автентичності відправника



Bob

* Приймає криптограму

** Розшифровує її та розкриває пару (ID_A, M)

*** Якщо прийнятий ідентифікатор ID_A збігається зі збереженим $ID`_A$ Bob визнає цю криптограму

2-й варіант безперервної перевірки автентичності відправника

Користувач Alice

Користувач Bob

2-й варіант безперервної перевірки відправника

- замість ідентифікатора відправника використовується його таємний пароль.
- Заздалегідь підготовлені паролі відомі обом сторонам.
- Нехай P_A і P_B – паролі користувачів Alice і Bob відповідно.

Alice створює криптограму

$$C = E_K(P_A, M)$$

Bob

* Розшифровує криптограму.

** Порівнює пароль, витягнутий з цієї криптограми, з початковим значенням.

*** Якщо паролі співпадають, то він визнає цю криптограму