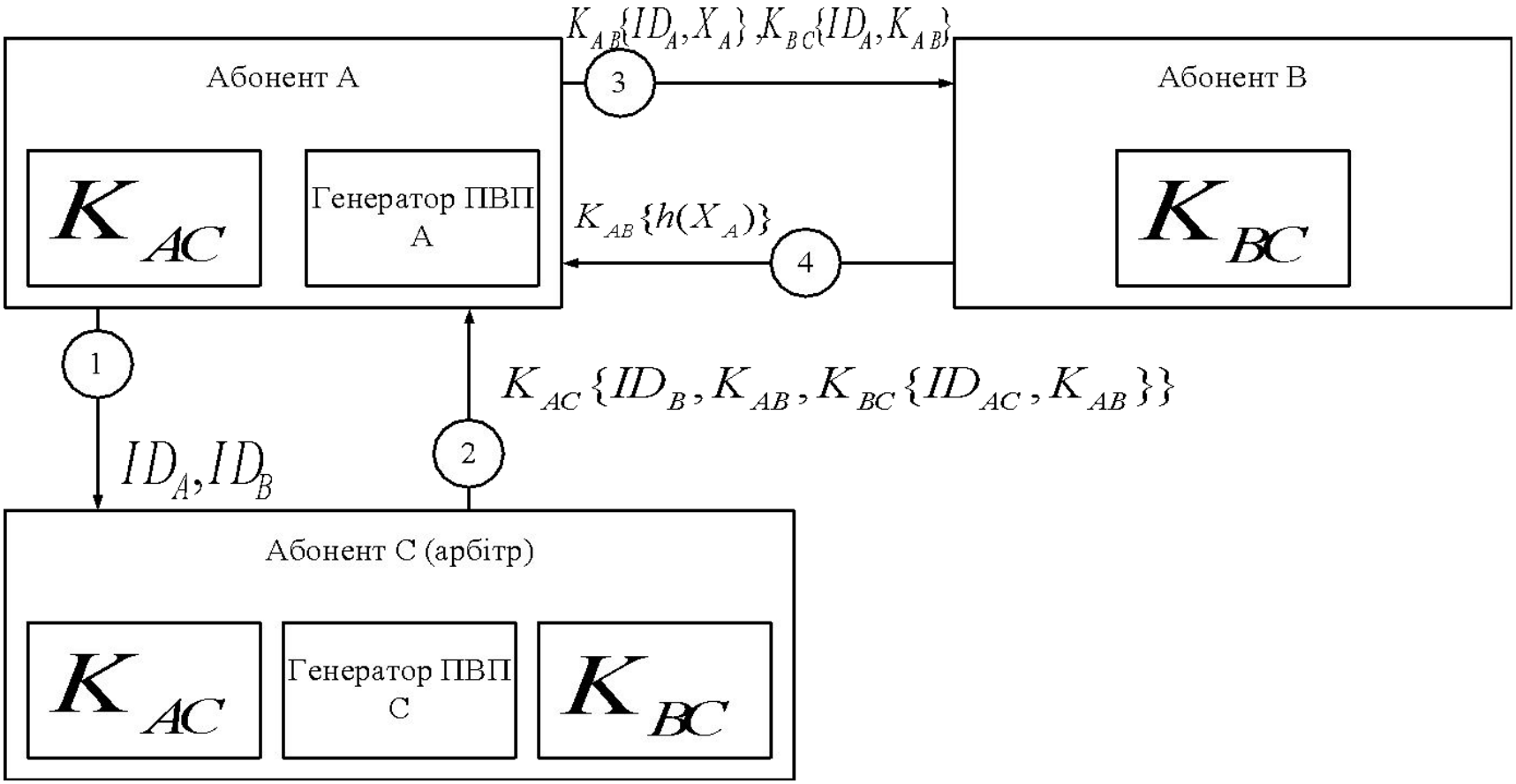
Захист інформації в
телекомунікаційних системах

ВЗАЄМНА АВТЕНТИФІКАЦІЇ СУБ'ЄКТІВ з довіреною третьою стороною

Лекція № 11

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

1 Схема симетричної автентифікації з довіреною третьою мережею



Довірена третя сторона (арбітр) Chris

- володіє секретними ключами K_{AC} і K_{BC} відповідно для взаємодії з Alice і Bob.

Крок 1: Alice бажає взаємодіяти з Bob та надсилає Chris

- повідомлення, що містить ідентифікатори суб'єктів взаємодії, що запитується ID_A та ID_B.

Крок 2: Chris отримавши повідомлення

формує сеансовий ключ K_{AB} для взаємодії суб'єктів Alice і Bob і посилає Alice зашифроване повідомлення

$K_{AC}\{ID_B, K_{AB}, K_{BC}\{ID_B K_{AB}\}\}$

Це зашифроване повідомлення

містить сеансовий ключ для роботи з Bob і шифровку, яка по суті є дозволом для Alice на роботу з Bob.

Крок 3: Alice розшифрувавши отримане повідомлення, визначає ключ K_{AB} і дозвіл

$K_{BC}\{ID_A, K_{AB}\}$

який вона розшифрувати не може, оскільки не знає ключа K_{BC}

4 Після цього Bob формує відповідь на запит і відправляє Alice повідомлення

$K_{AB}\{ID_A, h(x_A)\}$

Крок 5: Alice отримавши повідомлення

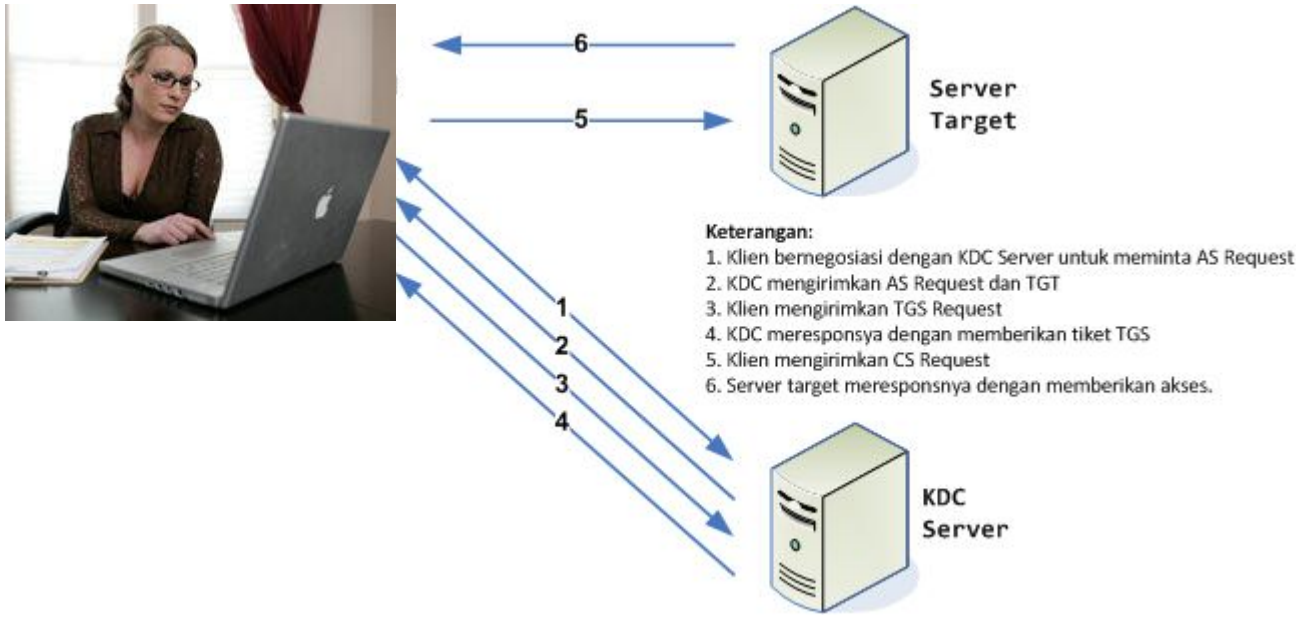
розшифровує його і перевіряє відповідь Bob;

у разі позитивного результату перевірки процес автентифікації успішно завершується.

2 Протокол Керберос

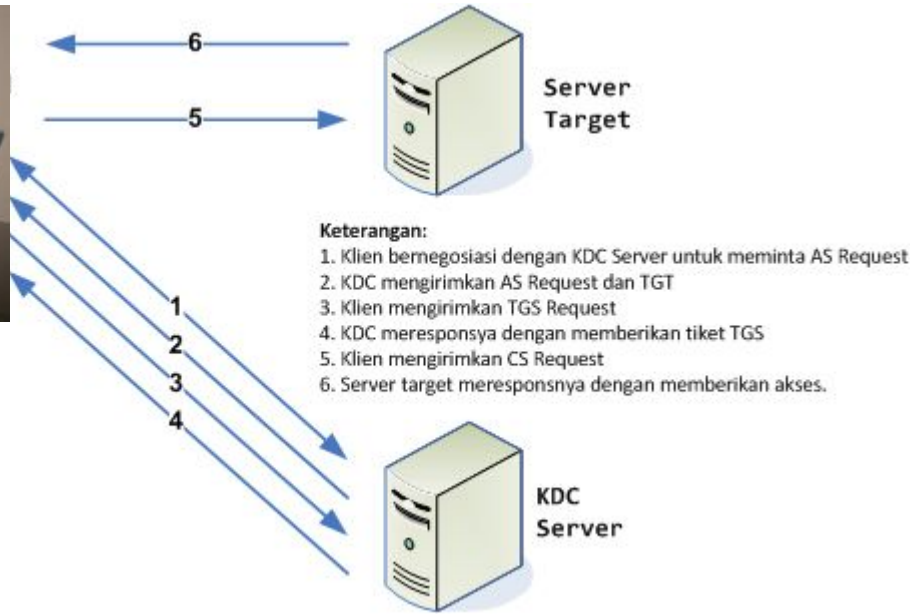
Схема передбачає взаємодію між трьома програмними компонентами

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



Kerberos

- Сервер Kerberos має доступ до бази даних, яка містить ідентифікатори і секретні ключі об'єктів.



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

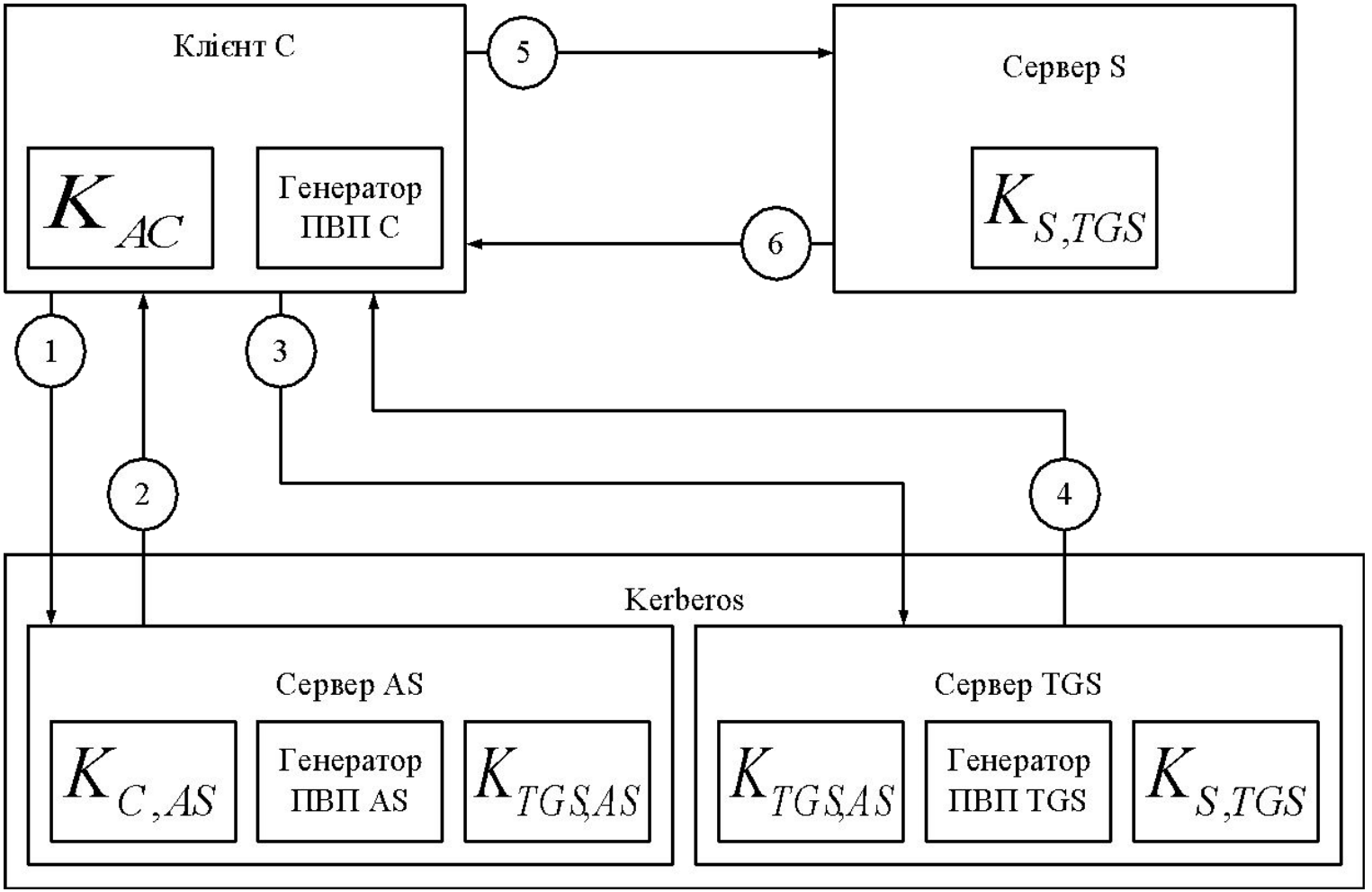
- Ідентифікатор суб'єкта;
- Секретний ключ суб'єкта;
- Дату закінчення терміну дії секретного ключа;
- Максимальний термін життя дозволів, які видаються суб'єкту;
- Номер версії секретного ключа суб'єкта;
- Дату останньої модифікації запису
- Іншу службову інформацію.

Програмне забезпечення сервера Kerberos

- розділене за своїми функціями на дві частини:

сервер автентифікації AS (Authentication Server)

сервер видачі дозволів (квитків) TGS (Ticket Granting Server).


Клієнт C
5
Сервер S
K_{AC}
Генератор ПВП C
6
$K_{S,TGS}$
1
3
2
4
Kerberos
Сервер AS
Сервер TGS
$K_{C,AS}$
Генератор ПВП AS
$K_{TGS,AS}$
$K_{TGS,AS}$
Генератор ПВП TGS
$K_{S,TGS}$

Крок 1 Клієнт С посилає серверу автентифікації повідомлення, що містить :

- ідентифікатор клієнта ID_C і ідентифікатор видачі дозволів ID_{TGS},
- інформацію , призначену для ідентифікації конкретного запиту:

  час,

  свій мережевий адрес та інш

Крок 2 – 1: Сервер автентифікації здійснює

- пошук в базі даних Kerberos за ідентифікатором клієнта і ідентифікатором послуги,
- знаходить відповідні ключі
- формує сеансовий ключ для взаємодії клієнта і сервера видачі дозволів

Крок 2-2: Після цього сервер AS посилає відповідь клієнту. Ця відповідь містить дві шифровки:

Перша, отримана на секретному ключі клієнта , містить:

сеансовий ключ для роботи з сервером видачі дозволів,

ідентифікатор клієнта,

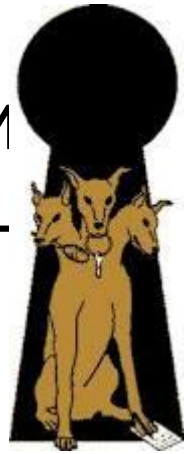
термін життя дозволу клієнта на роботу з сервером TGS.

Крок 2-3

- Друга шифровка, отримана на ключі , це дозвіл (ticket-granting ticket) клієнту на взаємодію з сервером TGS.
- До складу другої шифровки, яку клієнт прочитати не в змозі (оскільки не знає ключа) входять:
- ідентифікатори, сеансовий ключ і термін життя цього дозволу .

Крок 3

- Отримавши повідомлення, клієнт розшифровує першу його половину на ключі , перевіряє позначку , дізнається сеансовий ключ і термін життя дозволу на роботу з сервером TGS.
- В результаті обміну повідомленнями сервером AS, клієнт отримує дозвіл н роботу з сервером TGS.



Крок 3-2

- Потім клієнт посилає запит серверу видачі дозволів.
- Повідомлення для сервера TGS включає в себе дві шифровки.
- Перша, отримана на сеансовому ключі , включає в себе ідентифікатори , ідентифікатор запиту і тимчасову мітку .
- Друга – це «запечатаний» ключем дозвіл .

Крок 4 Сервер видачі дозволів

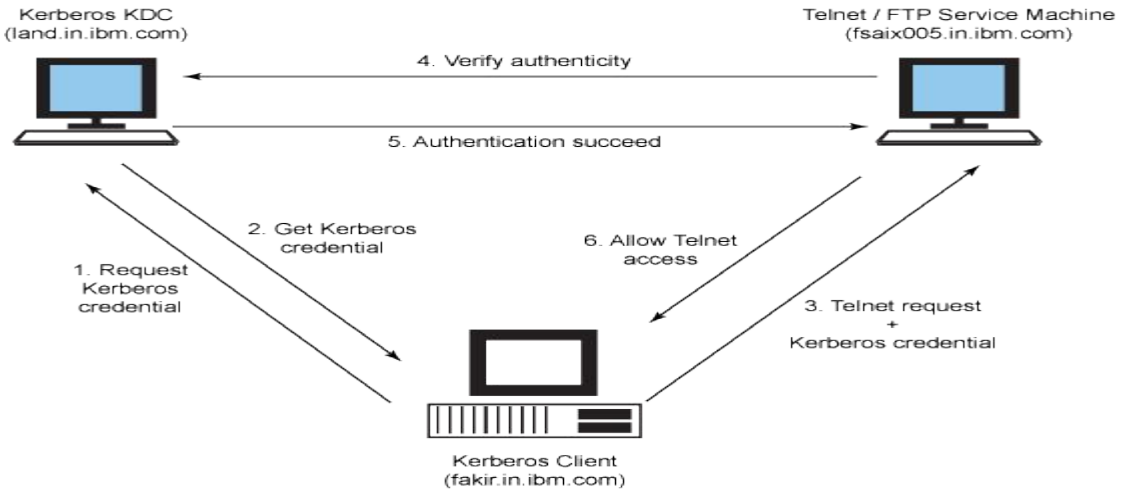
- розшифровує дозвіл
- дізнається сеансовий ключ , за допомогою якого читає першу частину прийнятого повідомлення,
- перевіряє ідентифікатор і тимчасову мітку .

Крок 4-2 Упевнившись у справжності клієнта, сервер TGS

- виробляє сеансовий ключ для взаємодії клієнта C і сервера S.
- На знанні цього ключа і ґрунтуватиметься в майбутньому взаємна автентифікація C і S.
- Після цього відправляється повідомлення клієнту, яке містить зашифровані на ключі , сеансовий ключ і термін життя дозволу клієнта на роботу з сервером, а також саме цей дозвіл , зашифрований на секретному ключі .

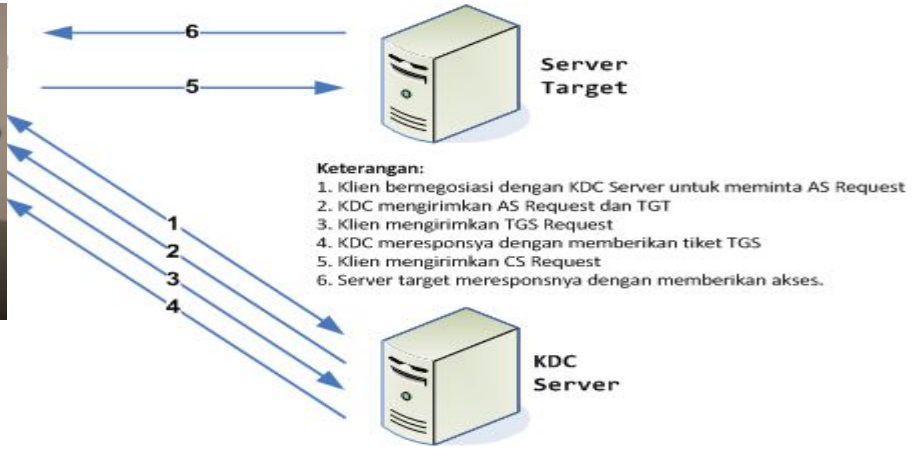
Крок 4-3 Після цього відправляється повідомлення клієнту, яке містить

- зашифровані на ключі , сеансовий ключ і термін життя дозволу клієнта на роботу з сервером,
- сам цей дозвіл , зашифрований на секретному ключі .



Крок 5-1

- Клієнт, отримавши повідомлення, розшифровує першу його частину з якої витягує сеансовий ключ для роботи з сервером S і термін життя дозволу на взаємодію з сервером S.



Крок 5-2

- Саме «запечатаний» ключем дозвіл клієнт прочитати не може.
- Таким чином, в результаті обміну з сервером видачі дозволів, клієнт отримує дозвіл на подальшу взаємодію вже з прикладним сервером.

Крок 5-3

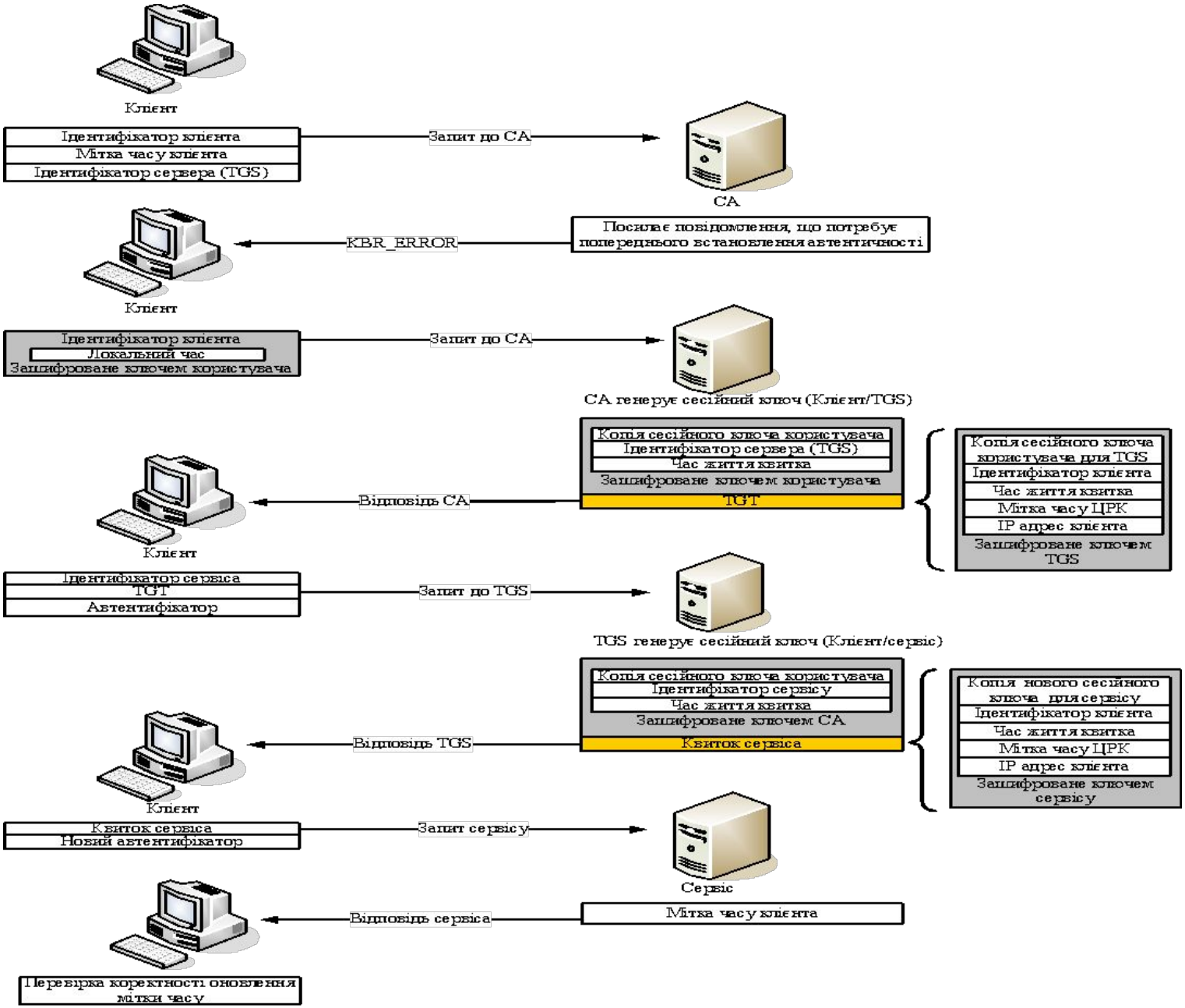
- Нарешті, клієнт посилає серверу S повідомлення, що містить свій ідентифікатор зашифрований на сеансовому ключі,
- тимчасову мітку
- також дозвіл , отриманий від сервера TGS.

Крок 6-1

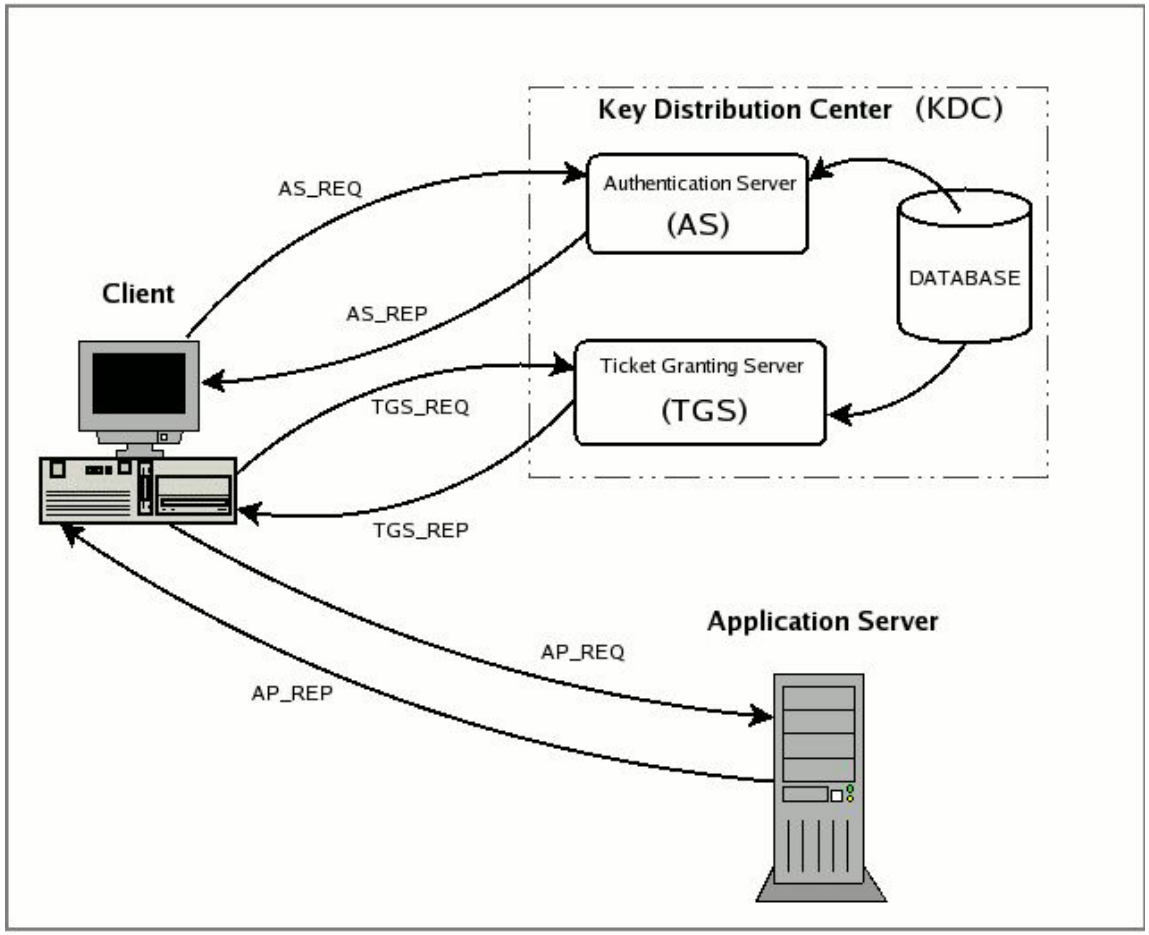
- Прийнявши повідомлення від клієнта і «розпечатавши» дозвіл , прикладний сервер дізнається сеансовий ключ і з його допомогою проводить автентифікацію клієнта, перевіряючи ідентифікатор і тим

Крок 6-2

- Відповідь сервера клієнту надсилається в тому випадку, коли потрібна взаємна автентифікація.
- Відповідь прикладного сервера в цьому випадку містить зашифрований на ключі результат перетворення мітки .


Клієнт
Ідентифікатор клієнта
Мітка часу клієнта
Ідентифікатор сервера (TGS)
Запит до CA
CA
Посилає повідомлення, що потребує попереднього встановлення автентичності
KBR_ERROR
Клієнт
Ідентифікатор клієнта
Локальний час
Зашифроване ключем користувача
Запит до CA
CA генерує сесійний ключ (Клієнт/TGS)
Копія сесійного ключа користувача
Ідентифікатор сервера (TGS)
Час життя квитка
Зашифроване ключем користувача
TGT
Копія сесійного ключа користувача для TGS
Ідентифікатор клієнта
Час життя квитка
Мітка часу ЦРК
IP адрес клієнта
Зашифроване ключем TGS
Відповідь CA
Клієнт
Ідентифікатор сервіса
TGT
Автентифікатор
Запит до TGS
TGS генерує сесійний ключ (Клієнт/сервіс)
Копія сесійного ключа користувача
Ідентифікатор сервісу
Час життя квитка
Зашифроване ключем CA
Квиток сервіса
Копія нового сесійного ключа для сервісу
Ідентифікатор клієнта
Час життя квитка
Мітка часу ЦРК
IP адрес клієнта
Зашифроване ключем сервісу
Відповідь TGS
Клієнт
Квиток сервіса
Новий автентифікатор
Запит сервісу
Сервіс
Мітка часу клієнта
Відповідь сервіса
Перевірка коректності оновлення мітки часу

3 Протокол Керберос 5



Крок 1

Клієнт C надсилає локальному серверу AS

Id_c , Id_{tgs} , N_c ,

N_c – величина, яка не повторюється в жодному сеансі протоколу, випадкове число

Крок 2

Локальний сервер AS надсилає клієнтові C

$$E_k(N_c, K_{c,as}, ID_{tgs}, T_{c,tgs})$$

$T_{c,tgs}$ – квітки для доступу для сервера видачі квитків

Крок 2 можна записати як

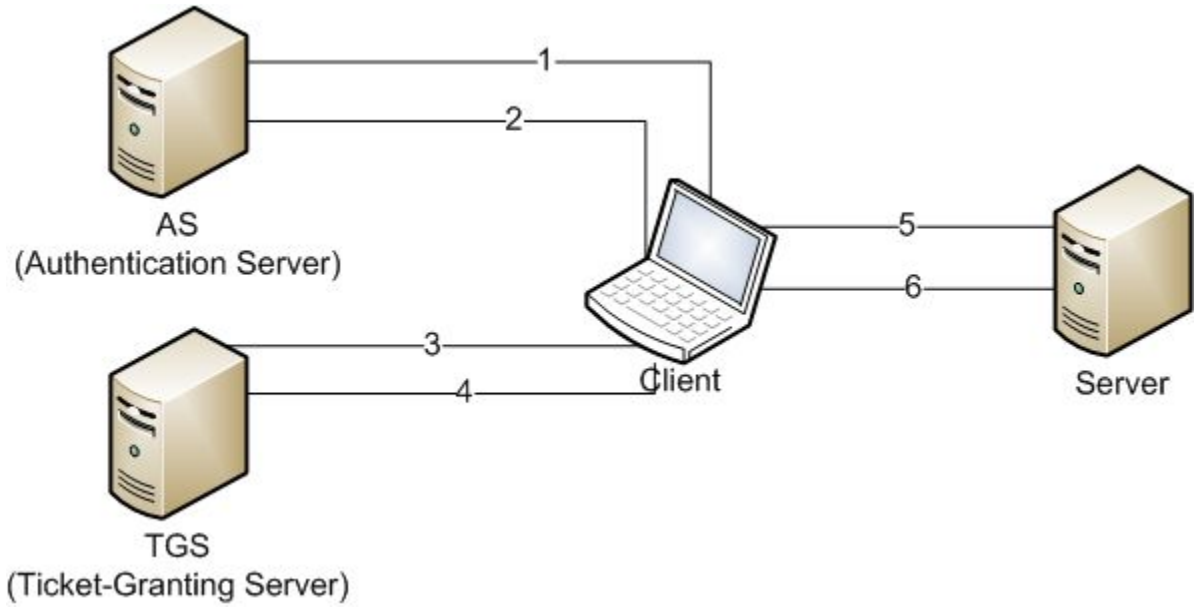
$\{ID_{tgs}, E_{kc,tgs}(Id_c, Id_{tgs}, t_{as}, l, K_{c,tgs})\}$

T_{as} – мітки часу

L – термін дії квитків

Кроки 1-2

Виконуються виключно під час першого входження клієнта до системи.



Крок 3

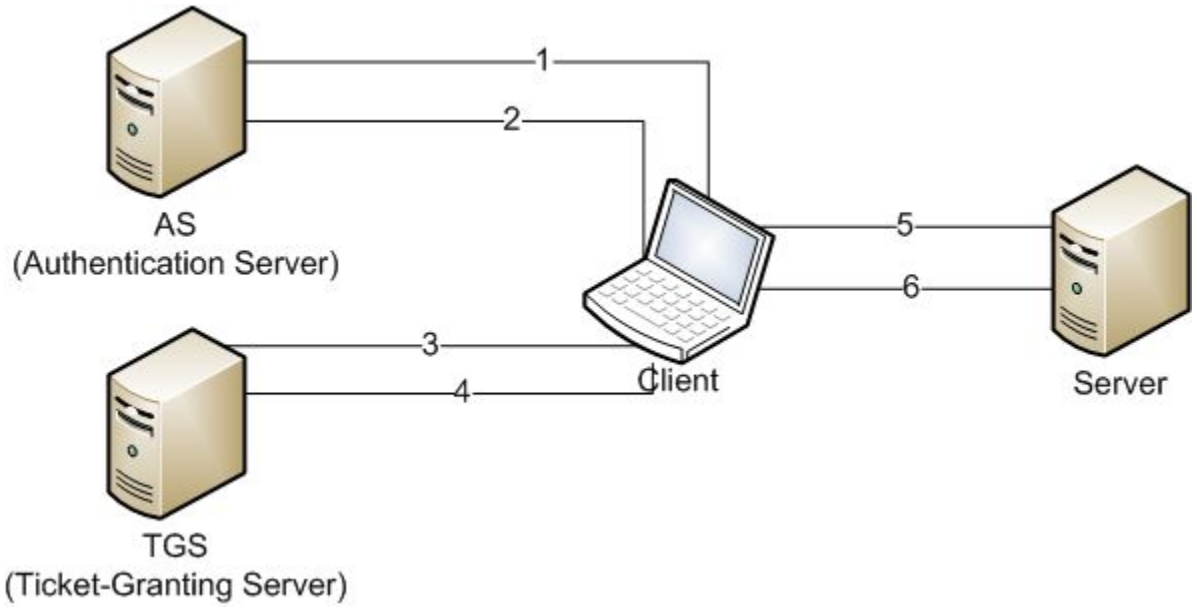
Клієнт С надсилає локальному серверу TGS автентифікатор для видачі квитків

$A_{c,tgs}=E_{kc,tgs}(ID_c, t_c\ldots)$,

ID_{tgs} –rem, $N_c, T_{c,tgs}$

Remark - REM

- Коментар # пояснювальний текст у програмі, макрокоманді або командному файлі. Ігнорується під час виконання



Крок 4

Локальний сервер TGP надсилає клієнтові С зашифроване повідомлення

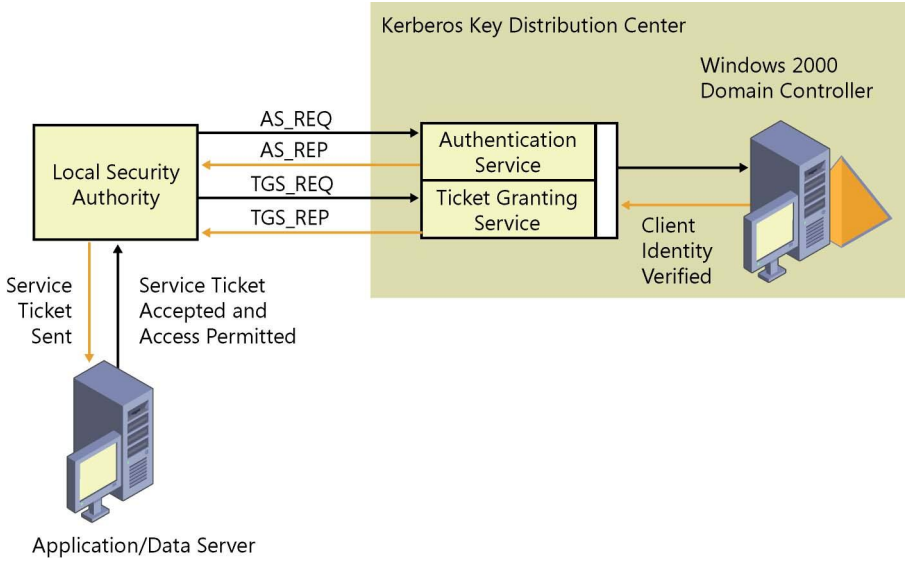
$$E_{kc,tgs}\ (K_{c,tgs},\ ID_{tgs\text{-}rem}\ -\ rem N_c,\ t_{ttgs},\ldots)$$

$$T_{c,tgs\text{-}\ rem},$$

де t_{ttgs} – мітка часу, $T_{c,tgs}$ – квіток до доступу

Кроки 3-4

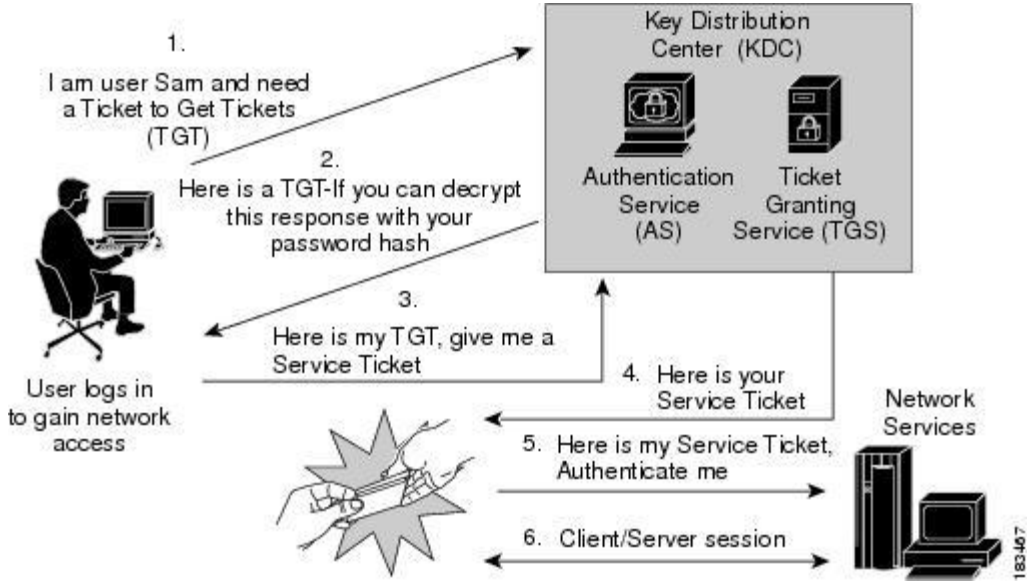
- Виконуються кожного разу коли клієнт C бажає звернутися до серверу S, що розташований у новому домені, до якого клієнт не звертався.



Крок 5

Клієнт C звертається до віддаленого сервера TGS:

$A_{c,tgs-rem}$, $T_{c,\ tgs-rem}$, IDs – rem,Nc

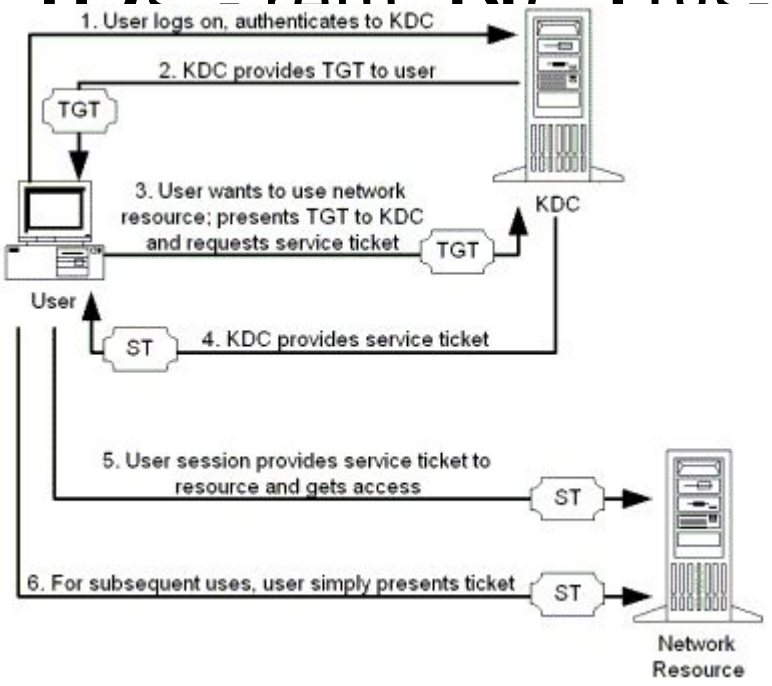


Крок 6

Віддалений TGS сервер надсилає клієнтові C:

$E_{kc,tgs\text{-}rem}$ ($K_{c,\ s\text{-}rem}$ $ID_{s\ \_\ rem}$ N_c $t_{tgs\text{-}rem}$)

$T_{c,s\text{-}rem}$



Кроки 5-6

- Виконуються кожного разу коли клієнт C звертається до нового сервера S у віддаленому відносно нього домені

Крок 7

- Клієнт C надсилає віддаленому серверу S

$A_{c.s\text{-}rem}$ =

$Ek_{c,s\text{-}rem}(ID_c, t_c, k_{c,s}$)

Крок 8

Віддалений сервер S надсилає клієнтові C

$A_{s\text{-}rem,c} =$

$$Ek_{c,s\text{-}rem}(IDs - rem, t_{c,+} L, k_{s\text{-}rem,c})$$

Крок 8

- Є необов`язковим та виконується, коли С вимагає від S взаємної автентифікації