МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
імені Семена Кузнеця

Лекція № 6

Змістовний модуль № 1: Системне програмування в Windows

по курсу "**Системне програмування**"

Тема лекції: Основи безпеки операційної системи

Лектор:
Доцент кафедри Інформаційних систем
кандидат технічних наук, доцент
Голубничий Дмитро Юрійович

04.10.2021

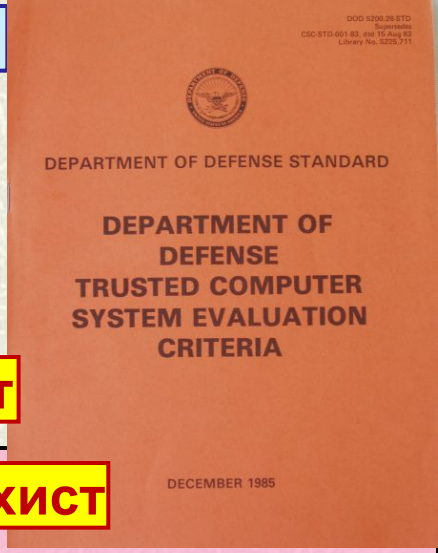
НАВЧАЛЬНІ ПИТАННЯ:

1.	Аутентифікація користувачів.
2.	Управління доступом.
2.1	Маркер доступу.
2.2	Ідентифікатор безпеки.
2.3	Привілеї .
2.4	Дескриптор безпеки .
2.5	Списки управління доступом.
3.	Інтерфейс CryptoAPI.

Вступ

Захищена багато користувачева система повинна:

- Захищати файли, пам'ять та інші ресурси кожного користувача від інших користувачів
- відслідковувати спроби обходу захисту
- захищати власні дані, файли і пам'ять від користувачевих програм



D Мінімальний захист

C Дискреційний захист

Рівень C2 Управління доступом

- забезпечувати контроль за доступом до ресурсів
- в момент входу в систему користувач повинен однозначно ідентифікувати себе
- система повинна запобігати себе від зовнішнього впливу або втручання в її роботу
- системний адміністратор повинен мати можливості перевіряти всі події, пов'язані з безпекою
- пам'ять повинна бути захищена

Рівень C1 Дискреційне забезпечення секретності

B Мандатний захист

- Рівень B3 Домени безпеки
- Рівень B2 Структурований захист
- Рівень B1 Захист із застосуванням мета-безпеки

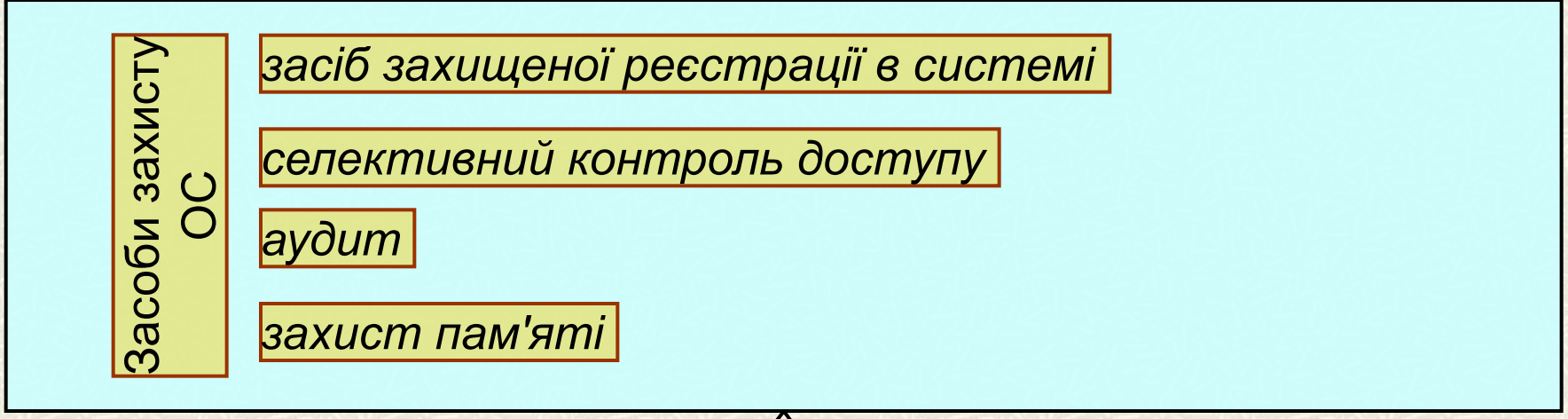
A Перевірений захист

- Рівень A1 Перевірений дизайн
- Рівень вище A1

Безпека

SD^3 + Communications
(by Design + Default + Deployment)

Безпека в архітектурі	▪безпечна архітектура ▪Новітні технології ▪Зменшення вразливості коду
Безпека за замовчуванням	▪Зменшення областей можливих атак ▪Відключення невикористаного за замовчуванням ▪Тільки мінімально необхідні привілеї
Безпека в роботі	▪Запобігання, виявлення, запобігання, відновлення, управління ▪навчання користувачів
взаємодія	▪Ясність поставлених цілей ▪Повноцінне членство в світовому співтоваристві ▪Microsoft Security Response Center



***Головна ідея**, лежить в основі системи безпеки Windows - це створення шлюзу, через який повинен пройти кожен користувач системних ресурсів*

аутентифікація і ідентифікація користувачів

управління доступом користувачів до об'єктів

1. Аутентифікація користувачів

СКОРОЧЕННЯ

LSA - Local Security Authority - **керуючий локальної безпекою**

SSPI - Security Support Provider Interface - **інтерфейс забезпечення безпеки**

SSP - security support provider - **провайдер підтримки безпеки**

LUID - locally unique identifier - **локальний унікальний ідентифікатор**

SAS - secure attention sequence - **комбінація CTRL + ALT + DEL**

SID - security identifier **ідентифікатори безпеки**

PT - primary token - **первинний маркер доступу**

RT - restricted token - **обмеженим маркером доступу**

ACL - access control list - **список контролю доступу**

ACE - access-control entries - **елементи контролю доступу**

DACL - discretionary access-control list - **список розмежувальної контролю доступу**

SACL - system access-control list - **системний список контролю доступу**

CSP - Cryptographic Service Provider - **криптопровайдер**

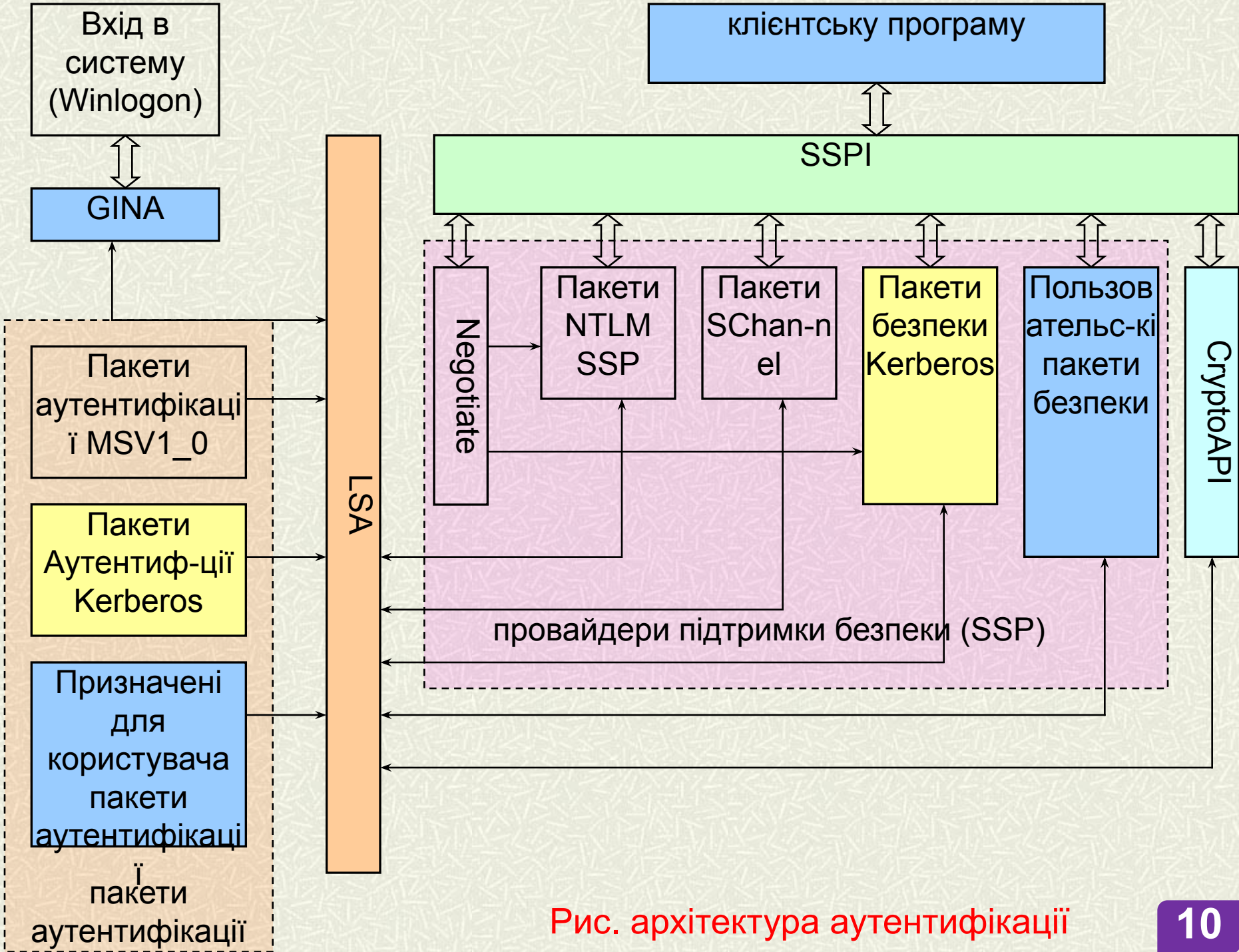
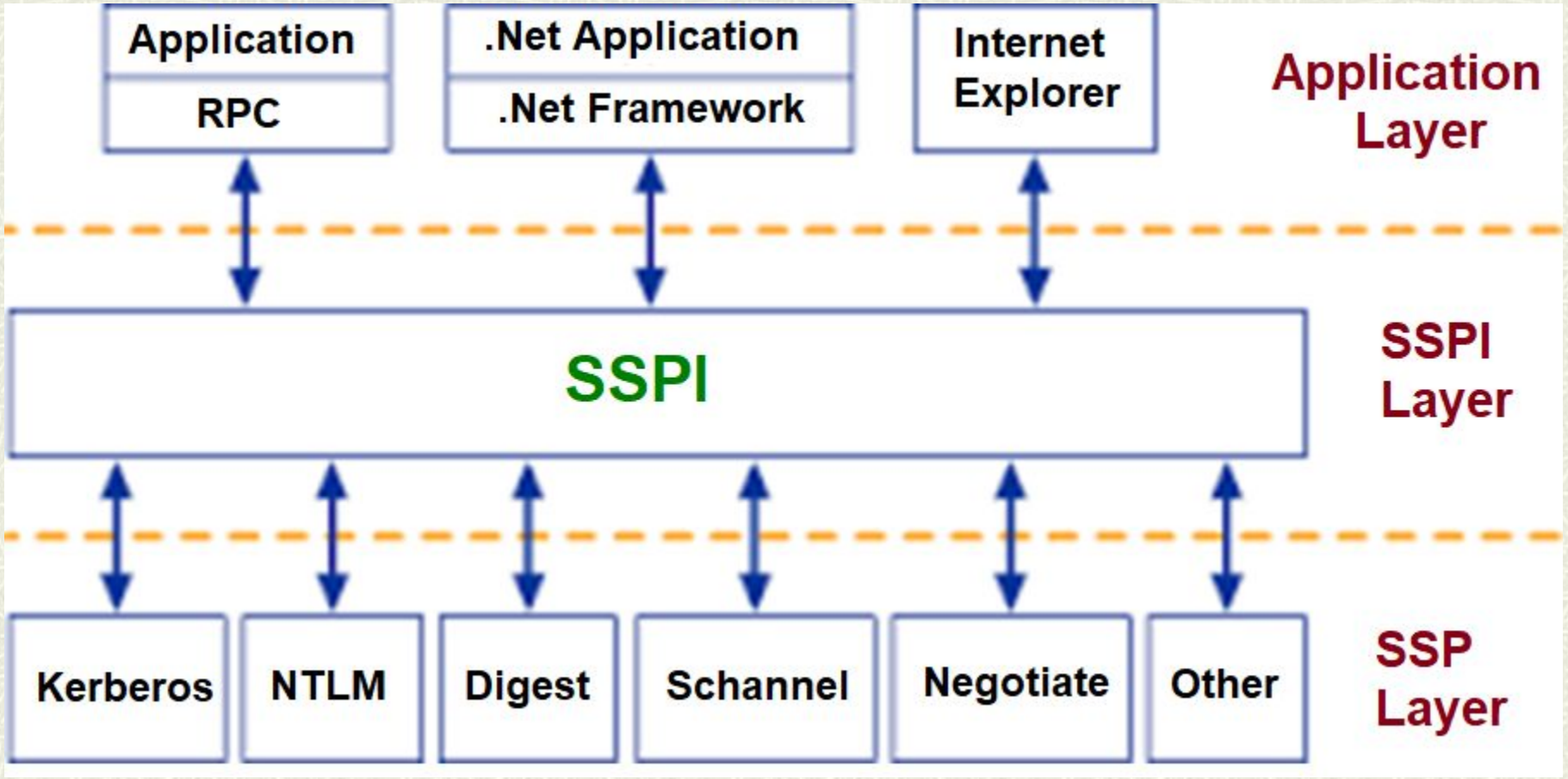
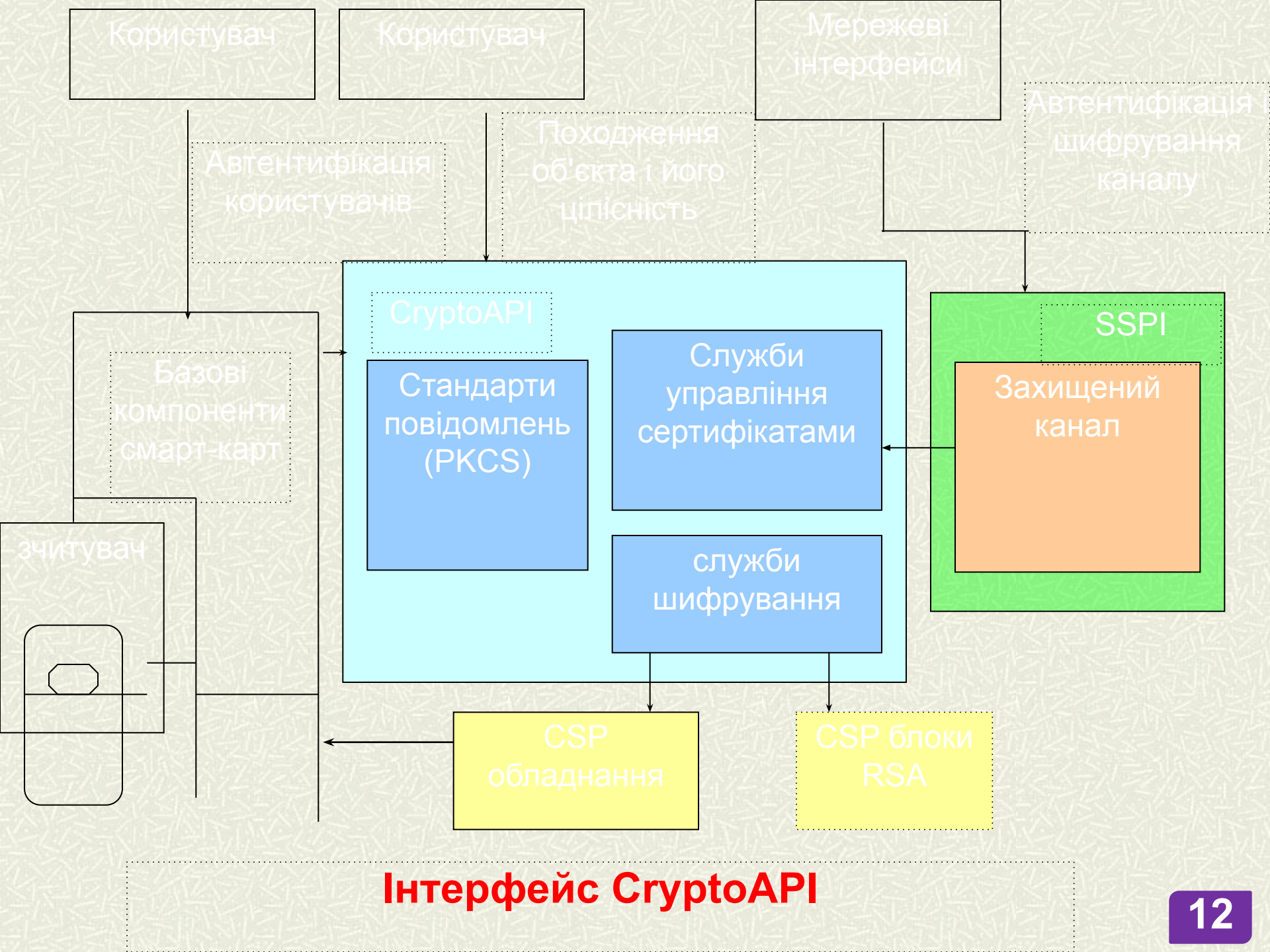


Рис. архітектура аутентифікації

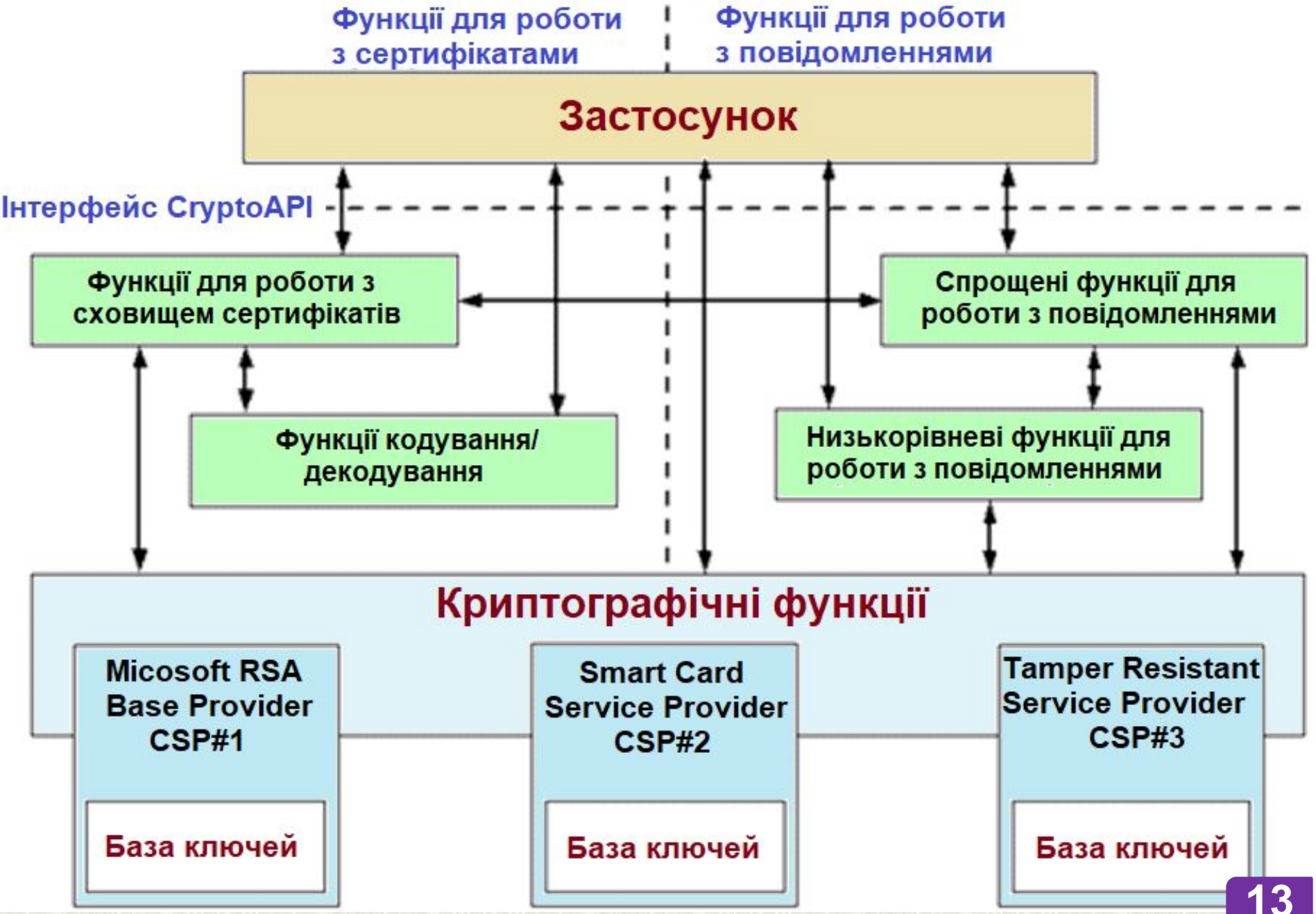
Проста діаграма інтерфейсу постачальника підтримки безпеки





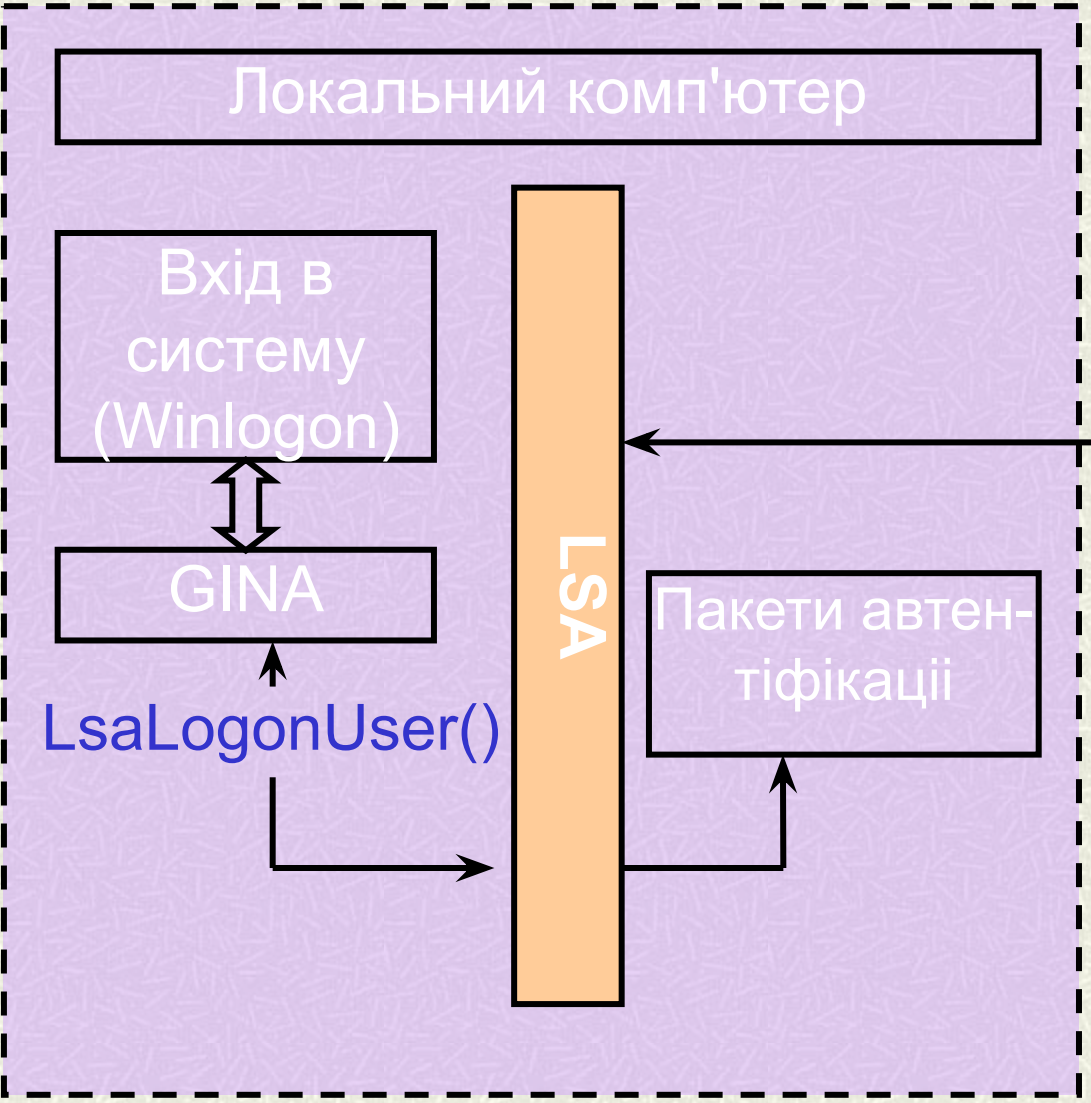
Інтерфейс CryptoAPI

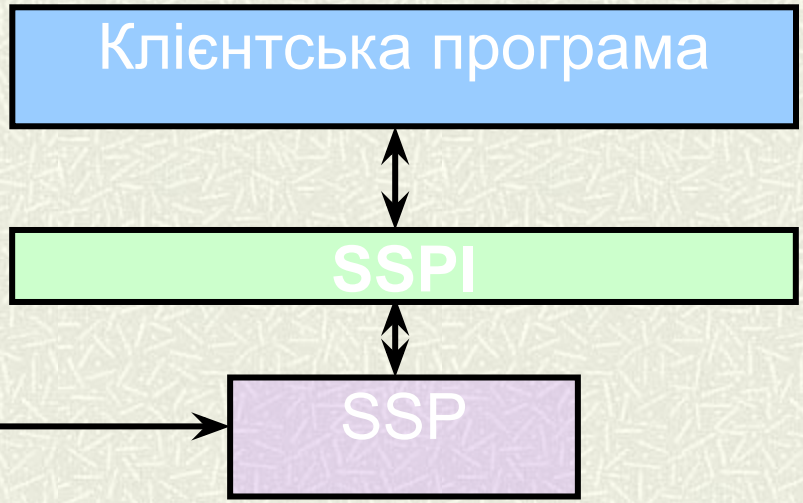
Архітектура CryptoAPI



Інтерактивна автентифікація

Не інтерактивна автентифікація





Можливості моделі **LSA** автентифікації

1. Модель LSA автентифікації підтримує зовнішні пакети автентифікації
2. В Windows LSA підтримує зовнішні пакети безпеки
3. LSA підтримує управління гетерогенними посвідченнями для взаємодії зі сторонніми продуктами

2. Управління доступом

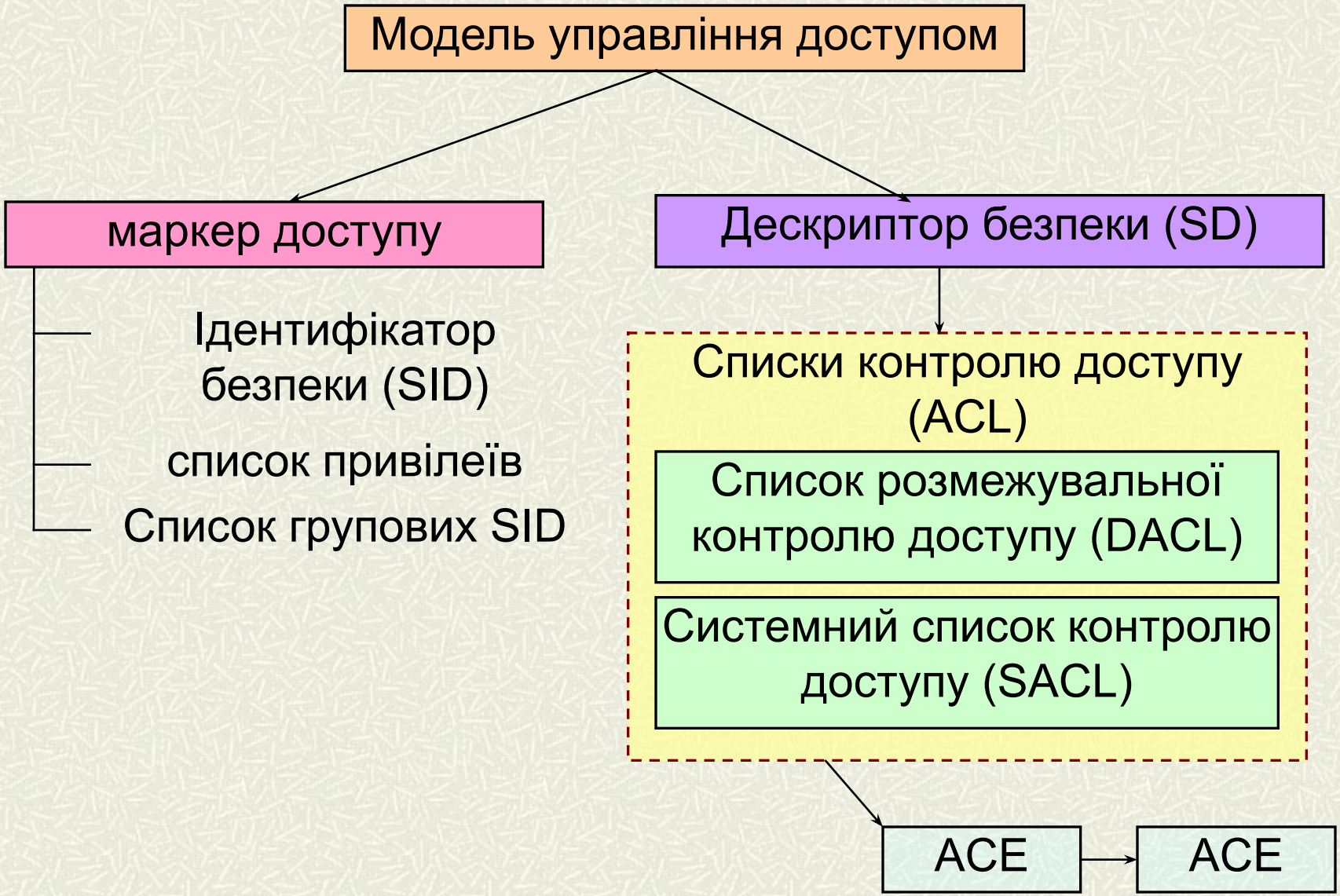


Рис.4. Модель управління доступом

2.1. Маркер доступу

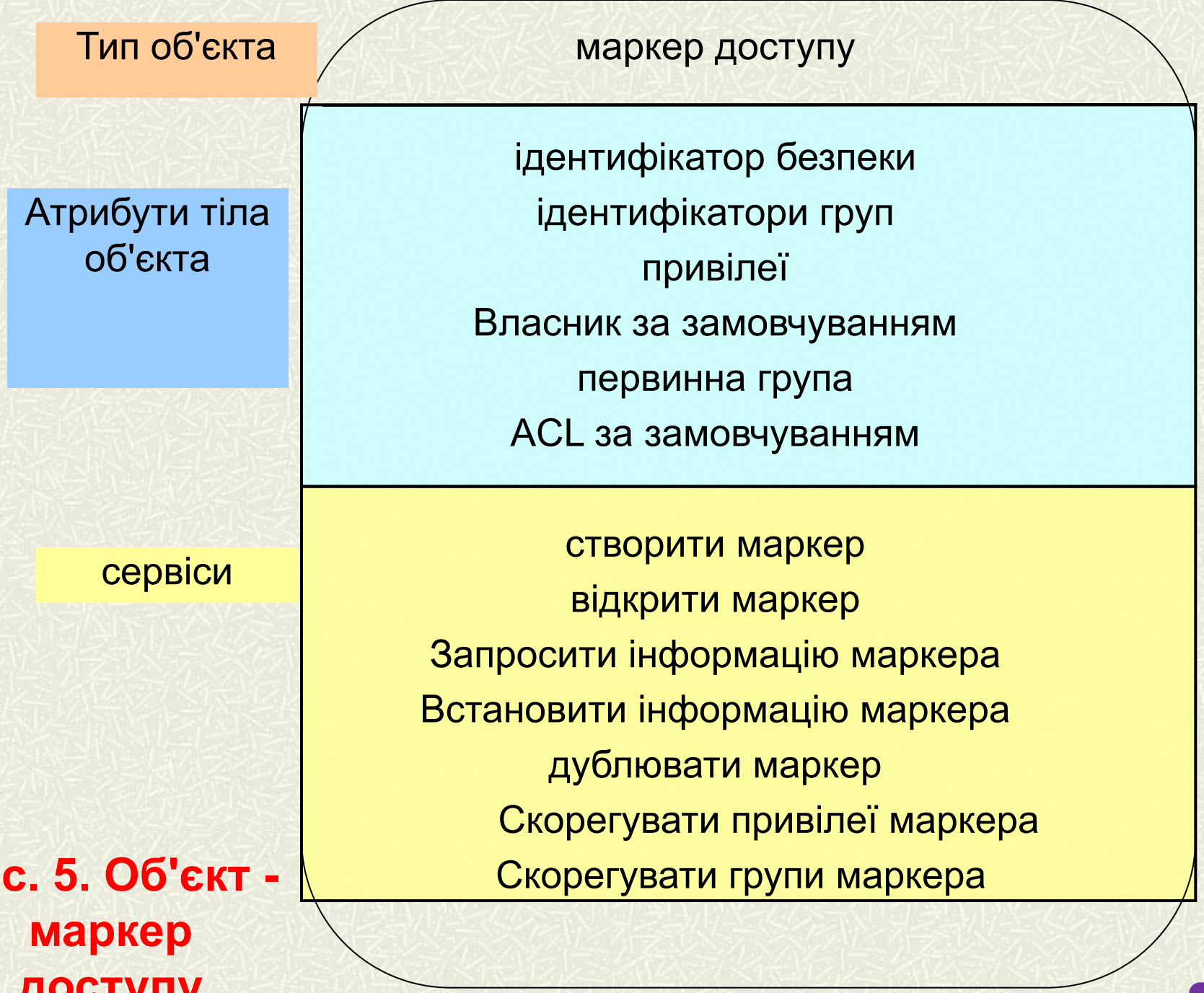


Рис. 5. Об'єкт - маркер доступу

Ідентифікатор безпеки: *IVANOVA*

Ідентифікатори груп: *TEAM1*

ADMINS

WORLD

привілеї: *немає*

Власник за замовчуванням: *IVANOVA*

Первинна група: *TEAM1*

ACL за замовчуванням:

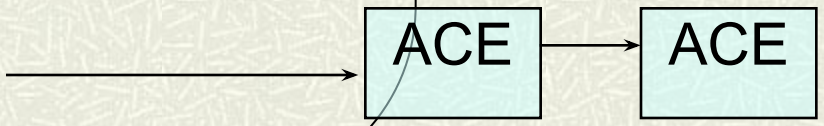


Рис. 6. Приклад маркера доступу

ТИПИ МАРКЕРІВ ДОСТУПУ

первинний маркер доступу
(primary token)

втілений маркер доступу
(impersonation token)

обмежений маркер доступу
(restricted token)

об'єкти, що захищають

Файли і каталоги файлової системи NTFS.

Іменовані канали.

Анонімні канали.

Процеси.

Потоки.

Об'єкти віртуальної пам'яті

маркери доступу

Об'єкти window station і desktop

ключі реєстру

Windows-сервіси

Локальні і віддалені принтери

Об'єкти синхронізації процесів

Об'єкти-завдання (job objects)

Мережеві колективні ресурси

Об'єкти служби каталогів

об'єкти, що захищають - це об'єкти, доступ до яких контролі-ється системою

Рис.7. об'єкти, що захищають

2.2. Ідентифікатор безпеки

Ідентифікатор безпеки (SID) - це структура змінної довжини, яка однозначно визначає користувача або групу користувачів

лістинг 1. структура SID

```
typedef struct _SID
{
BYTE Revision; // рівень перегляду SID
BYTE SubAuthorityCount; // кількість значень //
поставторізації
SID_IDENTIFIER_AUTHORITY IdentifierAuthority;/
*ідентифікатор авторизації* /
#ifdef MIDL_PASS
[Size_is (SubAuthorityCount)] DWORD SubAuthority [*];
# Else // MIDL PASS
DWORD SubAuthority [ANYSIZE_ARRAY];
#endif // MIDL_PASS
}SID, *PISID;
```

формат SID

S-*R*-*I*-*S*-*S*.....

рівень перегляду

ідентифікатор авторизації

рівні поставторізації

рівні поставторізації

48-біт 32-біт

ідентифікатор комп'ютера	***відносний ідентифікатор***

наприклад **S**-1-3256-81 .....

SID:

з рівнем перегляду = 1

рівнем поставторізації = 81

ідентифікатором авторизації = 3256

Функції для роботи з ідентифікаторами безпеки

ініціалізація структури SID

BOOL InitializeSid (PSID *pSid*, PSID_IDENTIFIER_AUTHORITY *pIdentifierAuthority*, BYTE *nSubAuthorityCount*)

Не тільки ініціалізація структури, а й покладання на систему турботу про виділення для неї пам'яті

BOOL AllocateAndInitializeSid
(PSID_IDENTIFIER_AUTHORITY
***pIdentifierAuthority*,**
BYTE *nSubAuthorityCount*,
DWORD *nSubAuthority0*, рівні поставторізації
DWORD *nSubAuthority1*,
DWORD *nSubAuthority2*,
DWORD *nSubAuthority3*,
DWORD *nSubAuthority4*,
DWORD *nSubAuthority5*,
DWORD *nSubAuthority6*,
DWORD *nSubAuthority7*,
PSID * pSid)

Функції для роботи з ідентифікаторами безпеки

отримання SID по імені користувача

BOOL LookupAccountName (LPCSTR *lpSystemName*, LPCSTR *lpAccountName*, PSID *Sid*, LPDWORD *cbSid*, LPSTR *ReferencedDomainName*, LPDWORD *cbReferencedDomainName*, PSID_NAME_USE *peUse*)

лістинг 2. структура SID_NAME_USE

```
typedef enum _SID_NAME_USE
{
SidTypeUser = 1,
SidTypeGroup,
SidTypeDomain,
SidTypeAlias,
SidTypeWellKnownGroup,
SidTypeDeletedAccount,
SidTypeInvalid,
SidTypeUnknown
} SID_NAME_USE, * PSID_NAME_USE;
```

Функції для роботи з ідентифікаторами безпеки

Розмір буфера, виділеного для SID для всіх рівнів поставторізації

DWORD **GetSidLengthRequired** (UCHAR *nSubAuthorityCount*)

Для визначення кількості рівнів поставторізації

PUCHAR **GetSidSubAuthorityCount** (PSID *pSid*)

Отримання імені користувача по SID

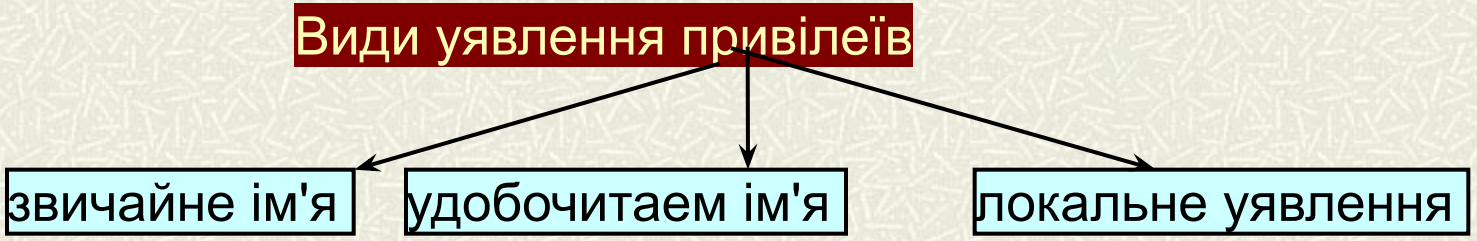
BOOL **LookupAccountSid** (LPCSTR *lpSystemName*, PSID *Sid*, LPSTR *Name*, LPDWORD *cbName*, LPSTR *ReferencedDomainName*, LPDWORD *cbReferencedDomainName*, PSID_NAME_USE *peUse)*

видалення SID

PVOID **FreeSid** (PSID pSid)

2.3. Привілеї

привілеївикористовуються для того, щоб більш строго контролювати доступ до ресурсів системи. Адміністратор мережі використовує привілеї для того, щоб визначати, хто з користувачів має право маніпулювати системними ресурсами. Додатки використовують привілеї в тих випадках, коли їм необхідно змінити системні ресурси



Наприклад, привілеї в Windows

привілей	*значення*	*призначення*
SE_MACHINE_ACCOUNT_NAME	TEXT ("SeMachineAccountPrivilege")	Отримання облікового запису комп'ютера
SE_LOAD_DRIVER_NAME	TEXT ("SeLoadDriverPrivilege")	Завантажити або вивантажити системний *драйвер*
SE_CREATE_PAGEFILE_NAME	TEXT ("SeCreatePagefile Privilege")	Створити сторінковий файл
SE_SHUTDOWN_NAME	TEXT ("SeShutdown Privilege")	Здійснювати вимикання локального комп'ютера

2.4. Дескриптор безпеки

Категорії об'єктів, які можуть мати дескриптор безпеки

- файли
- об'єкти користувача (вікна, зображення, кольору кисті і т.д.)
- об'єкти ядра
- об'єкти, визначені користувачем
- об'єкти реєстру

Лістинг 3. Структура дескриптора безпеки

```
typedef struct _SECURITY_DESCRIPTOR
{
BYTE Revision;
BYTE Sbz1;
SECURITY_DESCRIPTOR_CONTROL Control;
PSID Owner;
PSID Group;
PACL Sacl;
PACL Dacl; } SECURITY_DESCRIPTOR, * PISECURITY_DESCRIPTOR;
```

Об'єкт - файл

Список управління доступом (ACL)

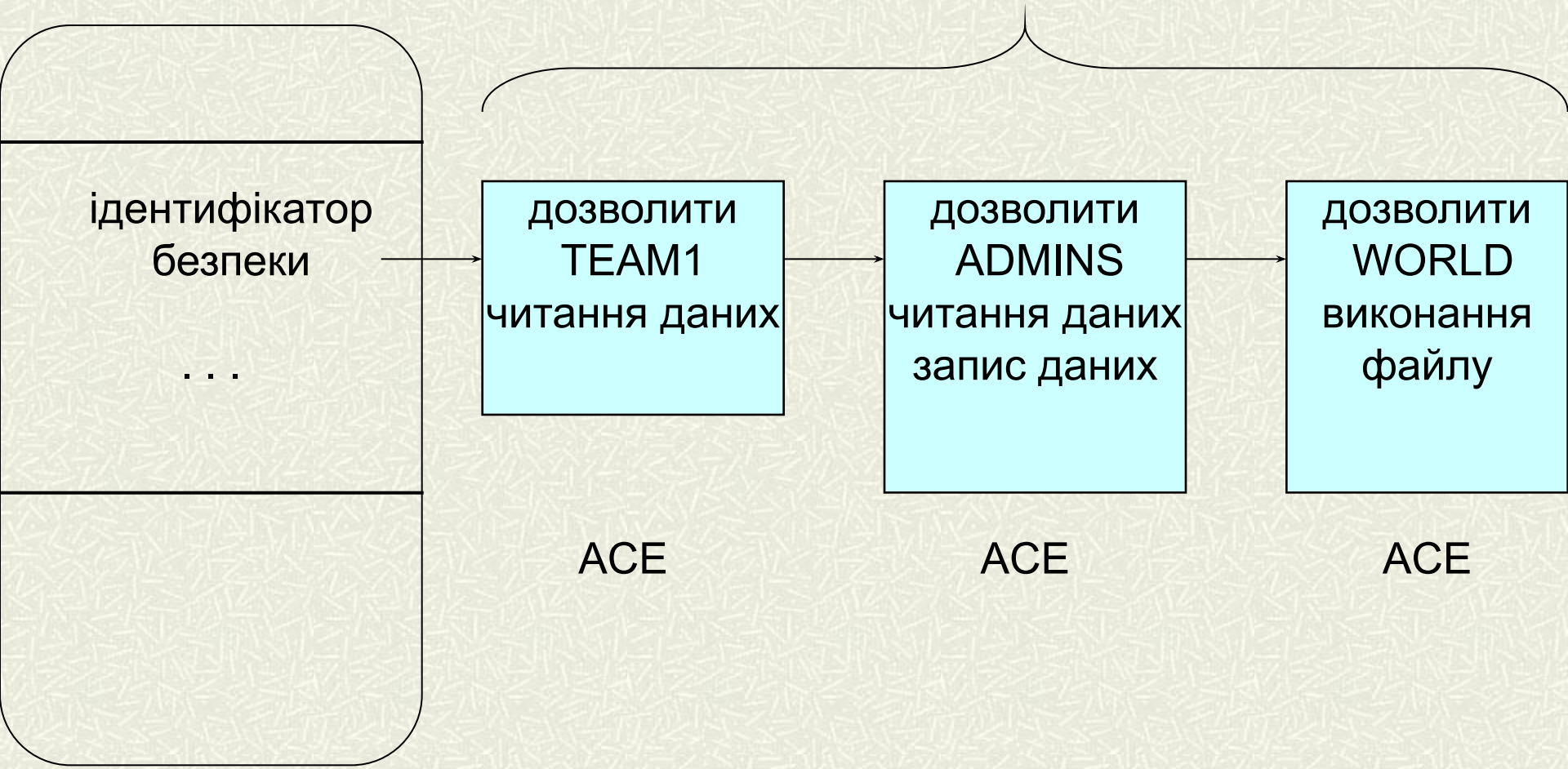


Рис. 8. Список управління доступом

Функції для роботи з дескриптором безпеки

WINADVAPI BOOL WINAPI GetFileSecurity (
LPCSTR *lpFileName*,
SECURITY_INFORMATION *RequestedInformation*,
PSECURITY_DESCRIPTOR *pSecurityDescriptor*, DWORD *nLength*,
LPDWORD *lpnLengthNeeded*)

прапори запиту інформації про безпеку

прапор	*призначення*
OWNER_SECURITY_INFORMATION	Запитується ідентифікатор власника об'єкта
GROUP_SECURITY_INFORMATION	Запитується ідентифікатор первинної групи об'єкта
DACL_SECURITY_INFORMATION	Запитується інформація про вільний ACL
SACL_SECURITY_INFORMATION	Запитується інформація про системний ACL

WINADVAPI BOOL WINAPI SetFileSecurity (LPCSTR *lpFileName*,
SECURITY_INFORMATION *SecurityInformation*, PSECURITY_DESCRIPTOR
***pSecurityDescriptor*)**

Функції для роботи з дескриптором безпеки

WINADVAPI BOOL WINAPI GetKernelObjectSecurity (
HANDLE *Handle*, SECURITY_INFORMATION *RequestedInformation*,
PSECURITY_DESCRIPTOR *pSecurityDescriptor*, DWORD *nLength*, LPDWORD *lpnLengthNeeded*).

WINADVAPI BOOL WINAPI SetKernelObjectSecurity (HANDLE *Handle*,
SECURITY_INFORMATION *SecurityInformation*, PSECURITY_DESCRIPTOR *SecurityDescriptor*).

WINUSERAPI BOOL WINAPI GetUserObjectSecurity (HANDLE *hObj*,
PSECURITY_INFORMATION *pSIRequested*, PSECURITY_DESCRIPTOR *pSID*,
DWORD *nLength*, LPDWORD *lpnLengthNeeded*).

WINUSERAPI BOOL WINAPI SetUserObjectSecurity (HANDLE *hObj*,
PSECURITY_INFORMATION *pSIRequested*, PSECURITY_DESCRIPTOR *pSID*)

WINADVAPI LONG APIENTRY RegGetKeySecurity (HKEY *hKey*,
SECURITY_INFORMATION *SecurityInformation*, PSECURITY_DESCRIPTOR *pSecurityDescriptor*, LPDWORD *lpcbSecurityDescriptor*)

WINADVAPI LONG APIENTRY RegSetKeySecurity (HKEY *hKey*,
SECURITY_INFORMATION *SecurityInformation*, PSECURITYJDESCRIPTOR *pSecurityDescriptor*)

2.5. Списки управління доступом

Список контролю доступу (access-control list (**ACL**)) Являє собою список елементів контролю доступу (access-control entries (**ACE**)). кожен елемент**ACE** у списку **ACL** ідентифікує довірену особу (trustee) І задає його права доступу (дозволу, заборони і аудит).

DACL (discretionary access-control list) - список розмежувальної контролю доступу ідентифікує довірених осіб, яким дозволений або заборонений доступ до захищається.

SACL (system access-control list) - системний список контролю доступу дозволяє адміністраторам протоколювати спроби доступу до захищених об'єктів.

Лістинг 4. опис заголовка ACL

```
typedef struct _ACL
{BYTE AclRevision;
BYTE Sbz1;
WORD AclSize;
WORD AceCount;
WORD Sbz2;
} ACL; typedef ACL *
PACL;
```

Типи загальних ACE для всіх об'єктів, що захищаються

ACE заборони доступу	Використовується в списку **DACL** для заборони доступу заданому довіреній особі
ACE дозволу доступу	Використовується в списку **DACL** для дозволу доступу заданому довіреній особі
ACE системного аудиту	Використовується в списку **SACL** для генерації записи аудиту в ситуаціях, коли заданий довірена особа намагається реалізувати задані права доступу

Типи об'єктно-специфічних ACE, що захищаються

ACE заборони доступу	Використовується в списку **DACL** для заборони доступу заданому довіреній особі до властивості або набору властивостей об'єкта, а також обмеження спадкування **ACE**для заданого типу дочірнього об'єкта. використовує структуру**ACCESS_DENIED_OBJECT_ACE**
ACE дозволу доступу	Використовується в списку **DACL** для дозволу доступу заданому довіреній особі до властивості або набору властивостей об'єкта, а також дозволу успадкування **ACE**для заданого типу дочірнього об'єкта. використовує структуру**ACCESS_ALLOWED_-OBJECT_ACE**
ACE системного аудиту	Використовується в списку **SACL** для протоколювання спроб доступу заданого довіреної особи до властивості або набору властивостей об'єкта, а також успадкування **ACE**для заданого типу дочірнього об'єкта. використовує структуру**SYSTEM_AUDIT_OBJECT_ACE**

лістинг 5. структура заголовка *ACE*

```
typedef struct _ACE_HEADER
{BYTE AceType;
BYTE AceFlags;
WORD AceSize;
}      ACE_HEADER;      typedef      ACE_HEADER      *
PACE_HEADER;
```

Тип входу в структурі ACE

Тип	*призначення*
ACCESS_ALLOWED_ACE	Доступ дозволено
ACCESS_DENIED_ACE	Доступ заборонено
SYSTEM_AUDIT_ACE	системна перевірка
SYSTEM_ALARM_ACE	В даний час не використовується

Лістинг 6. структура заголовка ACCESS_ALLOWED_ACE

```
typedef struct _ACCESS_ALLOWED_ACE
{ACE_HEADER Header;
ACCESS_MASK Mask;
DWORD SidStart;
} ACCESS_ALLOWED_ACE;
```

лістинг 7. структура заголовка ACCESS_MASK

```
typedef struct _ACCESS_MASK
{
WORD SpecificRights;
BYTE StandardRights;
BYTE AccessSystemAcl: 1;
BYTE Reserved: 3;
BYTE GenericAll: 1;
BYTE GenericExecute: 1;
BYTE GenericWrite: 1;
BYTE GenericRead: 1;
} ACCESS_MASK;
typedef ACCESS "MASK *
PACCESS_MASK;
```

Лістинг 8. структура заголовка ACCESS_DENIED_ACE

```
typedef struct _ACCESS_DENIED_ACE
{ACE_HEADER Header;
ACCESS_MASK Mask;
DWORD SidStart;
} ACCESS_DENIED_ACE;
typedef ACCESS_DENIED_ACE * PACCESS_DENIED_ACE;
```

Лістинг 9. структура заголовка ACCESS_AUDIT_ACE

```
typedef struct _SYSTEM_AUDIT_ACE
{ACE_HEADER Header;
ACCESSJWASK Mask;
DWORD SidStart;
} SYSTEM_AUDIT_ACE;
```

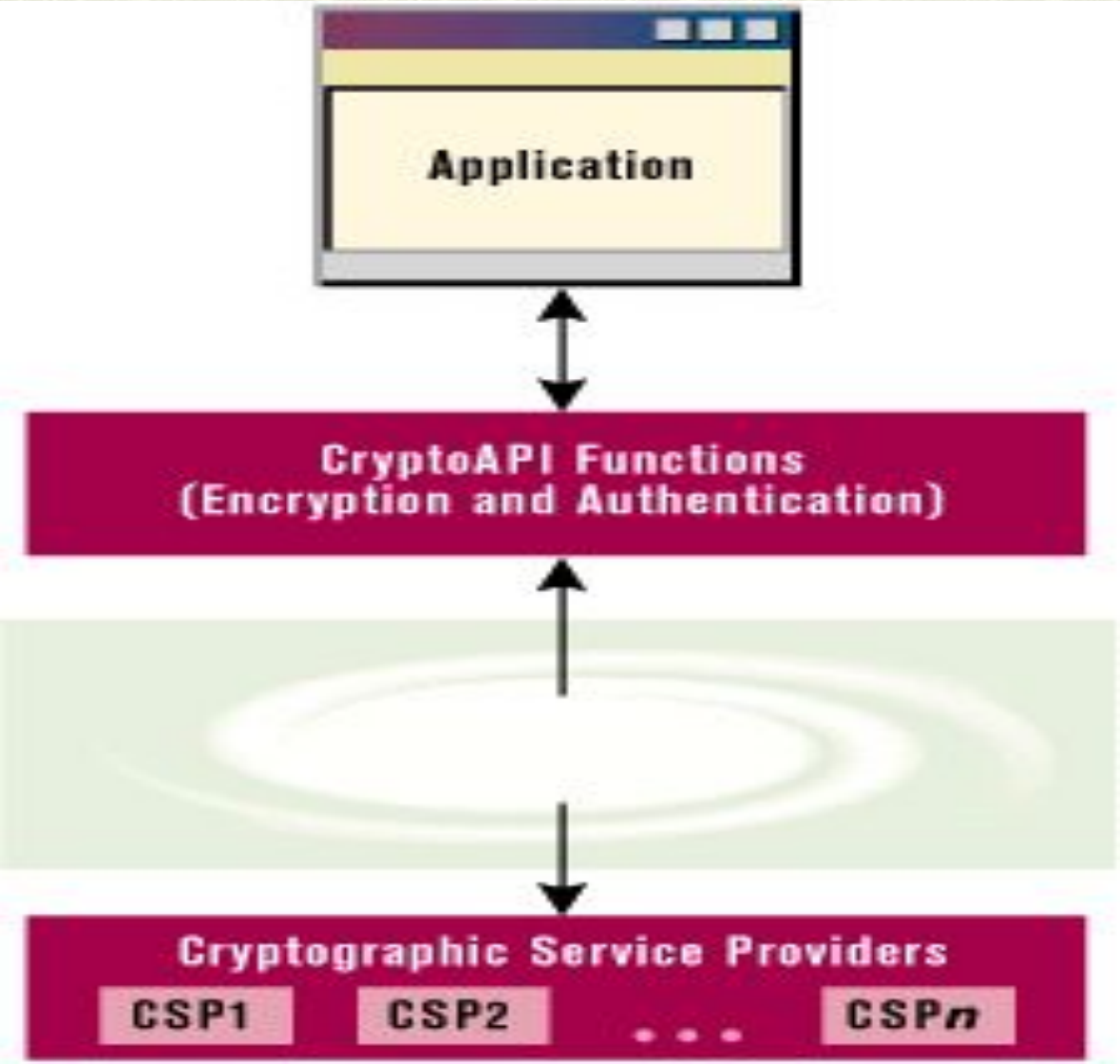
Функції для роботи з ACL

WINADVAPI BOOL WINAPI InitializeAcl (PACL *pAcl*, DWORD *nAclLength*, DWORD *dwAclRevision*);

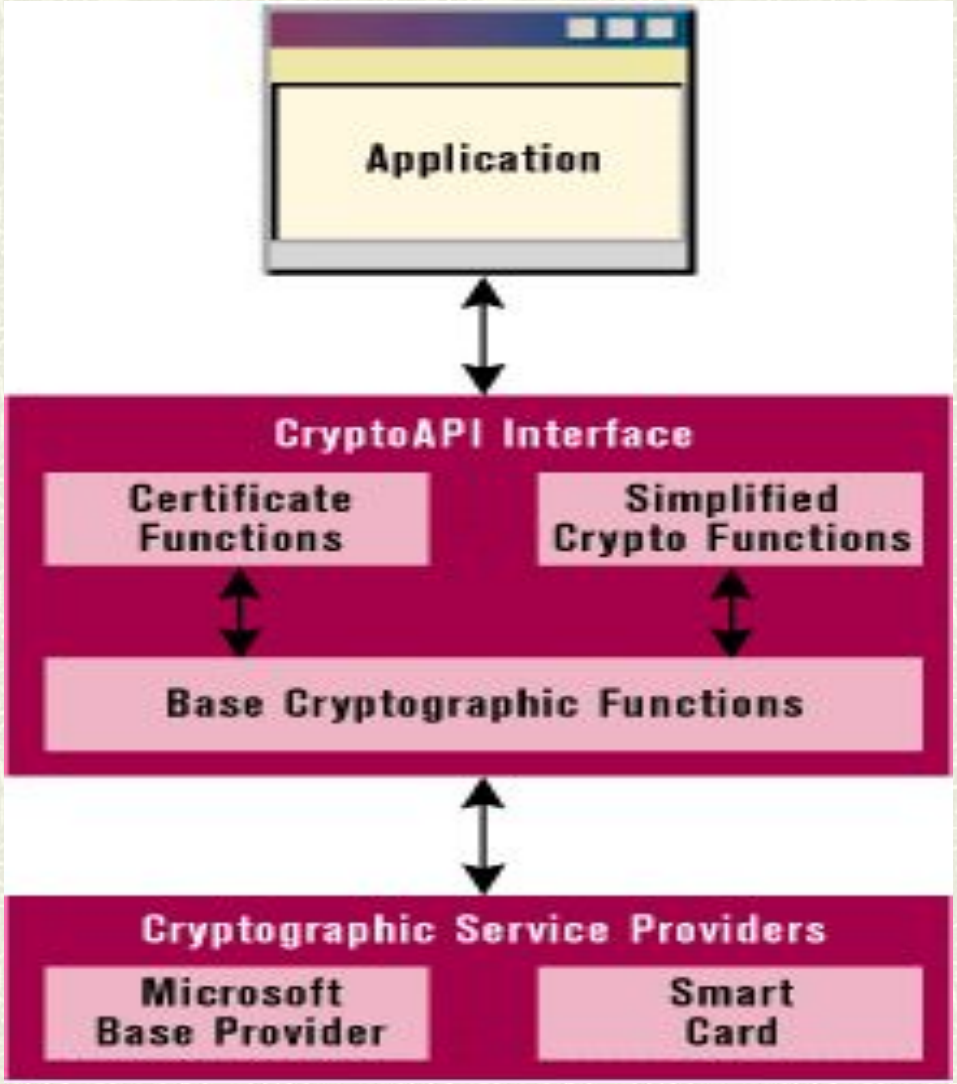
WINADVAPI BOOL WINAPI GetSecurityDescriplorDacl (PSECURITY_DESCRIPTOR *pSecurityDescriptor*, LPBOOL *lpbDaclPresent*, PACL **pDacl*, LPBOOL *lpbDaclDefaulted*);

WINADVAPI BOOL WINAPI GetSecurityDescriptorSacl (PSECURITY_DESCRIPTOR *pSecurityDescriptor*, LPBOOL *lpbSaclPresent*, PACL **pSacl*, LPBOOL *lpbSaclDefaulted*);

3. Інтерфейс CryptoAPI



Основна Модель CryptoAPI



Детальна Модель Crypto API

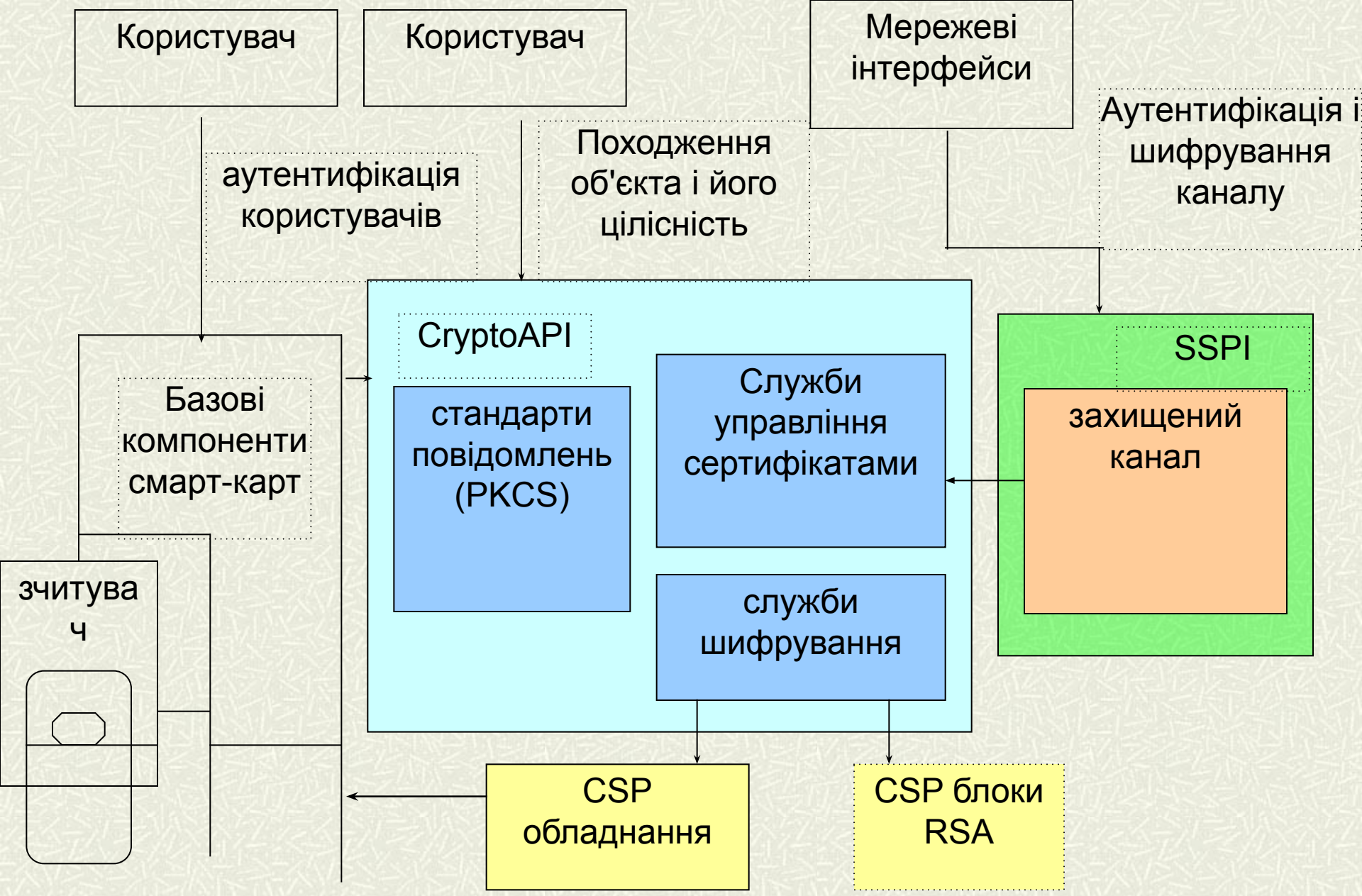


Рис. 11. Інтерфейс CryptoAPI

Функції забезпечення криптопровайдера

№ п / п	*функція*	*Короткий опис*
1.	CryptAcquireContext	Використовується для отримання дескриптора ключового контейнера всередині відповідного криптопровайдера
2.	CryptContextAddRef	Інкрементує лічильник посилань на HCRYPTPROV дескриптор
3.	CryptEnumProviders	Перераховує криптопровайдери, встановлені на комп'ютері
4.	CryptEnumProviderTypes	Перераховує типи криптопровайдерів, встановлених на комп'ютері
5.	CryptGetDefaultProvider	Визначає криптопровайдер, який використовується за замовчуванням, для поточного користувача або для комп'ютера
6.	CryptGetProvParam	Отримує параметри криптопровайдера
7.	CryptReleaseContext	Звільняє дескриптор криптопровайдера, отриманий через виклик функції CryptAcquireContext
8.	CryptSetProvider	Встановлює криптопровайдер, який використовується за замовчуванням, для відповідного типу криптопровайдера
9.	CryptSetProviderEx	
10.	CryptSetProvParam	Встановлює параметри криптопровайдера

Функції управління та передачі ключової інформації

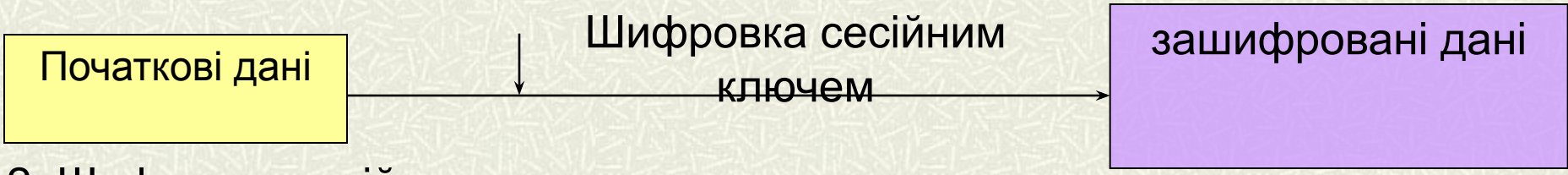
№ п / п	функція	Короткий опис
11	CryptDeriveKey	Створює ключ, що отримується з пароля
12	CryptDestroyKey	знищує ключ
13	CryptExportKey	Експортує ключ з криптопровайдера в ключовий блоб в пам'яті додатків
14	CryptGenKey	Створює випадковий ключ
15	CryptDuplicateKey	Робить точну копію ключа і його характеристик
16	CryptGenRandom	Генерує випадковий блок даних
17	CryptGetKeyParam	Отримує параметри ключа
18	CryptGetUserKey	Отримує дескриптор ключа обміну або електронного підпису
19	CryptImportKey	Імпортує ключ з ключового БЛОБ в криптопровайдер
20	CryptSetKeyParam	Встановлює параметри ключа

функції шифрування/ дешифрування даних

№ п / п	функція	Короткий опис
21	CryptDecrypt	Розшифровує фрагмент шифртекста, використовуючи зазначений ключ шифрування
22	CryptEncrypt	Шифрує розділ відкритого тексту, використовуючи зазначений ключ шифрування

```
BOOL CRYPTFUNC CryptEncrypt (
HCRYPTKEY   hKey,   //      дескриптор      ключа      для
шифрування
HCRYPTHASH hHash
BOOL   bFinal
BYTE * pbData,// параметр [In, out]
DWORD *   pdwDataLen,    // параметр [In, out]
DWORD     dwBufferLen)
```

1. Шифрування даних



2. Шифровка сесійного ключа

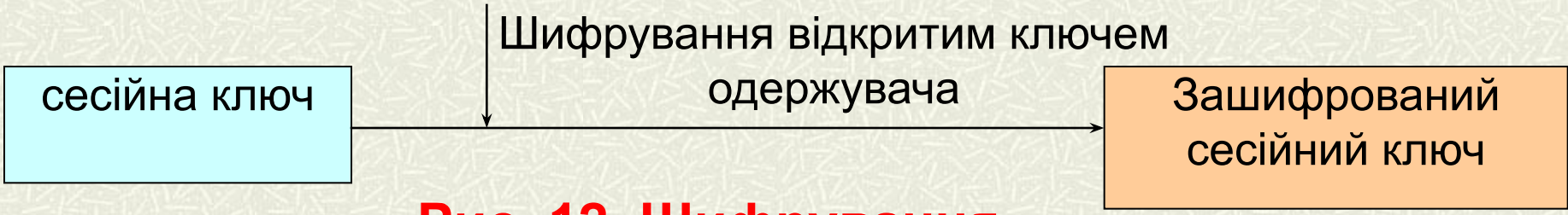
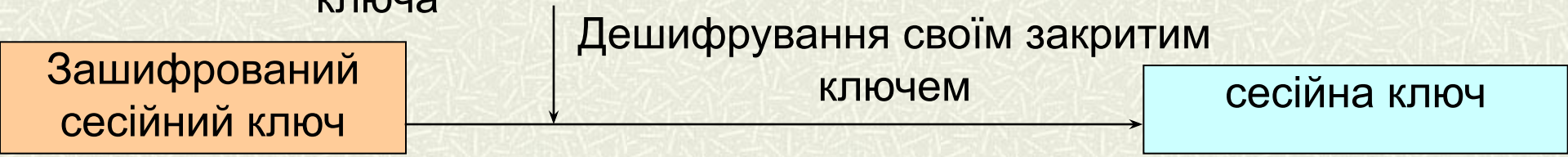


Рис. 12. Шифрування

1. Дешифровка сесійного ключа



2. Дешифрування вихідних даних

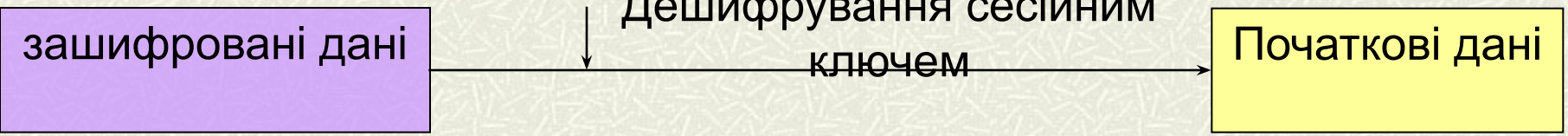
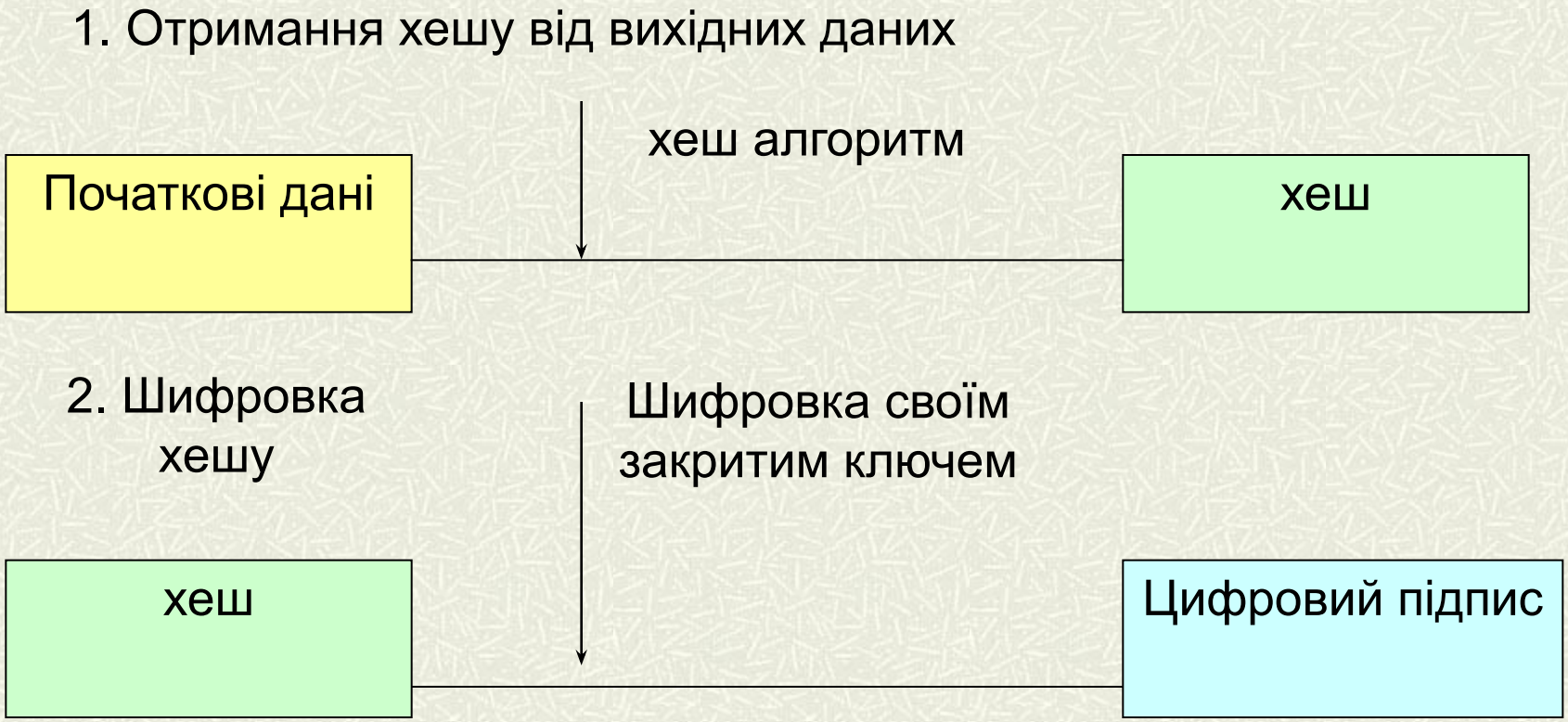


Рис. 13. Дешифрування

Функції хешування та електронного цифрового підпису

№ п / п	функція	Короткий опис
23	CryptCreateHash	Створює "порожній" об'єкт хешу
24	CryptDestroyHash	Знищує об'єкт хешу
25	CryptDuplicateHash	Дублює об'єкт хешу
26	CryptGetHashParam	Отримує параметри об'єкта хешу
27	CryptHashData	Хешірует блок даних, додаючи хеш до зазначеного об'єкту хешу
28	CryptHashSessionKey	Хешірует сесійний ключ і додає значення хешу до зазначеного об'єкту хешу
29	CryptSetHashParam	Встановлює параметри об'єкта хешу
30	CryptSignHash	Підписує вказаний об'єкт хешу
31	CryptVerifySignature	Перевіряє цифровий підпис

Цифровий підпис - це двійкові дані невеликого обсягу, зазвичай не більше 256 байт. Цифровий підпис є не що інше, як результат роботи хеш-алгоритму над вихідними даними, зашифрований закритим ключем відправника.



Створення цифрового підпису


Дякую за увагу!