ТОВАРИСТВО З ОБМЕЖЕНОЮ ВІДПОВІДАЛЬНІСТЮ
С.І.Т

ЗАТВЕРДЖЕНИЙ
UA.38773869.00002-11 13 02-А3

Засіб криптографічного захисту інформації «3DA CA Client»
Опис програми

Бібліотека UAPKI
Інтерфейс прикладного програмування

UA.38773869.00002-11 13 02

Київ – 2024

Загальні відомості

Цей документ призначено для використання розробниками додатків, які мають потреби у інтеграції механізмів електронного підпису та шифрування даних з використанням сертифікатів відкритих ключів.

Бібліотека UAPKI є компонентом засобу криптографічного захисту інформації «3DA CA Client» та безпосередньо реалізує функціонал пов'язаний з формуванням та перевірянням електронних підписів, зашифрування та розшифрування даних з використанням сертифікатів відкритих ключів та інші допоміжні функції. До її складу входять бінарні файли, які представлені в таблиці 1 (префікси і розширення залежать від операційної системи).

Бібліотека підтримує роботу з різними носіями ключової інофрмації (надалі – НКІ), що містять приватні ключі електронного підпису та протоколу узгодження ключів, у тому числі: файлами формату PKCS#12, апаратними та апаратно-програмними засобами створення кваліфікованих електронних підписів тощо.

Таблиця 1. Перелік бінарних файлів

№	Базова назва файлу	Опис
1	uapki	Головна бібліотека, що реалізує основну логіку роботи
2	uapkic	Бібліотека криптографічних примітивів, обов'язкова
3	uapkif	Бібліотека форматів даних та роботи з синтаксисом ASN.1, обов'язкова
4	cm-<storage-name>	Бібліотеки роботи з НКІ (надалі – проовайдери НКІ). Наприклад, для НКІ у вигляді файлу формату PKCS#12 це буде "cm-pkcs12". Необхідні у разі використовуються функції, які залежать від приватних ключів (наприклад, підпис даних)

Взаємодія з бібліотекою відбувається за допомогою виклику методів, які представлені в таблиці 2. Методи викликаються за допомогою двох експортованих функцій бібліотеки: process і json_free, інтерфейс яких описаний в таблиці 3. Вся взаємодія з методами бібліотеки базується на використанні текстової строки, що складається за правилами JSON (надалі – JSON-строка).

Бібліотека підтримує багатопотоковістть, тобто можливість одночасного паралельного виклику методів з різних потоків. За особливостями реалізації в бібліотеці, методи поділяються на три типи підтримки багатопотоковості (тип підтримки багатопотоковості вказано у таблиці 2):

- незалежні (НЗ), не змінюють внутрішній стан бібліотеки, не блокують інші методи та не блокуються іншими методами;
- однопотокові (ОП), змінюють внутрішній стан бібліотеки, можуть виконуватися тільки в один потік, при виклику блокують всі інші однопотокові та багатопотокові методи;
- багатопотокові (БП), змінюють внутрішній стан бібліотеки, при виклику можуть блокувати інші потоки під час доступу до спільних ресурсів.

Таблиця 2. Перелік методів бібліотеки

№	Назва методу	Короткий опис	Тип
1	VERSION	Версія бібліотеки	НЗ
2	INIT	Ініціалізація бібліотеки	ОП
3	DEINIT	Завершення роботи бібліотеки (де-ініціалізація)	ОП
4	PROVIDERS	Перелік провайдерів НКІ	ОП
5	STORAGES	Перелік НКІ доступних через обраний провайдер	ОП
6	STORAGE_INFO	Інформація про НКІ	ОП
7	OPEN	Відкрити НКІ	ОП
8	CLOSE	Закрити НКІ	ОП
9	KEYS	Перелік ключів у відкритому НКІ	ОП
10	SELECT_KEY	Вибрати ключ	ОП
11	CREATE_KEY	Створити ключ	ОП
12	DELETE_KEY	Видалити ключ	ОП
13	GET_CSR	Отримати запит на сертифікат	ОП
14	CHANGE_PASSWORD	Зміна пароля (PIN коду) до НКІ	ОП
15	INIT_KEY_USAGE	Ініціалізація використання ключа	ОП
16	SIGN	Підпис даних	БП
17	VERIFY	Перевірка підписаних даних	БП
18	ENCRYPT	Шифрування даних	БП
19	DECRYPT	Розшифрування даних	БП
20	ADD_CERT	Додати сертифікат до кешу сертифікатів	БП
21	CERT_INFO	Інформація про сертифікат	БП
22	GET_CERT	Отримати сертифікат із кешу сертифікатів	БП
23	LIST_CERTS	Перелік сертифікатів у кеші сертифікатів	БП
24	REMOVE_CERT	Видалити сертифікат із кешу сертифікатів	ОП
25	VERIFY_CERT	Валідація сертифікату	БП
26	CERT_STATUS_BY_OCSP	Формування OCSP-запиту та отримання OCSP-відповіді	БП
27	ADD_CRL	Додати СВС до кешу СВС	БП
28	CRL_INFO	Інформація про СВС	БП
29	LIST_CRLS	Перелік СВС у кеші СВС	БП
30	REMOVE_CRL	Видалити застарілі СВС із кешу СВС	ОП
31	RANDOM_BYTES	Генерація псевдовипадкової послідовності	БП
32	DIGEST	Гешування даних	НЗ
33	ASN1_DECODE	Декодування DER-кодованих ASN.1 даних	НЗ
34	ASN1_ENCODE	Кодування даних згідно DER-кодування ASN.1	НЗ

Таблиця 3. Перелік експортованих функцій бібліотеки

№	Назва функції	Короткий опис
1	process	char* process(const char* request);
2	json_free	void json_free(char* result);

Функція **process**

Функція призначенна для виклику методу бібліотеки, передачі йому параметрів та отримання результату його виконання.

Вхідні параметри:
request – вказівник на нуль-терміновану JSON-строку в кодуванні UTF8, яка визначає метод, що викликається та його параметри (назва методу, параметри методу).

Вихідні параметри:
повертає вказівник на нуль-терміновану JSON-строку в кодуванні UTF8, що містить результат виконання методу. Пам'ять, на яку посилається цей вказівник, після обробки має завжди звільнятися функцією json_free.

Функція **json_free**

Функція призначена для звільнення пам'яті, що була виділена функцією process для повернення результату виконання методу бібліотеки.

Вхідні параметри:
параметр result – це вказівник на нуль-терміновану JSON-строку в кодуванні UTF8, який був повернутий функцією process;

Вихідні параметри:
немає.

Опис методів бібліотеки

Всі запити і відповіді методів мають єдиний формат.

Формат запиту

Назва поля	**Тип**	**Опис**
method	String	Назва методу. Обов'язковий параметр
parameters	Object	Опціональний параметр. Структура, що містить вхідні параметри методу

Формат відповіді

Назва поля	**Тип**	**Опис**
errorCode	Integer	Код помилки. Коди помилок наведено в Додатку А
method	String	Назва методу
result	Object	Структура, що містить вихідні параметри методу
error	String	Короткий текстовий опис помилки. Опціональний

Метод VERSION

Метод призначено для визначення версії бібліотеки.
Вхідні параметри: відсутні.

Структура поля result у відповіді

Назва поля	Тип	Опис
name	String	Ім'я бібліотеки
version	String	Номер версії бібліотеки

Приклад запиту:

```
{
  "method": "VERSION"
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "VERSION",
  "result": {
    "name": "UAPKI",
    "version": "2.0.11"
  }
}
```

Метод INIT

Метод призначено для ініціалізації бібліотеки. Вхідні параметри опціональні. У вихідних параметрах повертаються поточні значення статусу/параметрів підсистем бібліотеки. Якщо бібліотека була ініціалізована, то перед завершенням роботи з нею потрібно виконати метод DEINIT. Перелік методів для яких ініціалізація бібліотеки не обов’язкова: VERSION, DIGEST, ASN1_DECODE та ASN1_ENCODE.

Параметри бібліотеки можна задати двома способами: параметрами або через файл конфігурації. Параметри та файл конфігурації мають однакову структуру. Якщо використовується файл конфігурації, то необхідно в поле "configFile" вказати шлях до нього (рекомендована назва "uapki-config.json").

Якщо параметри бібліотеки не задані — будуть використані параметри за замовченням. Для роботи з НКІ необхідно задати параметри провайдерів НКІ.

Структура поля parameters у запиті з використанням файлу конфігурації

Назва поля	Тип	Опис
configFile	String	Повний шлях до файлу

Приклад запиту з використанням файлу конфігурації:

```
{
  "method": "INIT",
  "parameters": {
    "configFile": "C:/uapki/uapki-config.json"
  }
}
```

Структура поля parameters у запиті

Назва поля	Тип	Опис
cmProviders	Object CMPROVIDERS_PARAMS	Параметри провайдерів НКІ. Опціональний
certCache	Object CERT_CACHE_PARAMS	Параметри кешу сертифікатів. Опціональний.
crlCache	Object CRL_CACHE_PARAMS	Параметри кешу СВС. Опціональний
offline	Boolean	Режим роботи “офлайн”. Опціональний
ocsp	Object OCSP_PARAMS	Параметри OCSP-сервісу. Опціональний
proxy	Object PROXY_PARAMS	Параметри PROXY-сервісу. Опціональний
tsp	Object TSP_PARAMS	Параметри TSP-сервісу. Опціональний

Структура CERT_CACHE_PARAMS

Назва поля	Тип	Опис
path	String	Повний шлях до каталогу. Опціональний. Якщо шлях до каталогу не заданий використовується тільки тимчасовий кеш сертифікатів в памяті
trustedCerts	Base64[]	Масив довірених сертифікатів. Опціональний

Структура CRL_CACHE_PARAMS

Назва поля	Тип	Опис
path	String	Повний шлях до каталогу. Опціональний. Якщо шлях до каталогу не заданий використовується тільки тимчасовий кеш СВС в памяті
useDeltaCrl	Boolean	Використовувати частковий СВС. Опціональний, за замовчанням true

Структура OCSP_PARAMS

Назва поля	Тип	Опис
nonceLen	Integer	Довжина одноразового випадкового числа в OCSP-запиті. Діапазон значень: 0, 8..64. Якщо значення дорівнює 0 або знаходиться не в діапазоні, то випадкове число в OCSP-запиті не використовується. Опціональний, за замовчанням дорівнює 20

Структура PROXY_PARAMS

Назва поля	Тип	Опис
url	String	URL-адреса PROXY-сервісу. Якщо значення відсутнє або є пустим рядком, то PROXY-сервіс не використовується. Опціональний.
credentials	String	Параметри автентифікації до PROXY-сервісу. Опціональний

Структура TSP_PARAMS

Назва поля	Тип	Опис
certReq	Boolean	Вимога повертати в TSP-відповіді сертифікат сервісу. Опціональний, за замовчанням дорівнює false
forced	Boolean	Вимога використовувати URL-адреси TSP-сервісів, які задані в полі "url", ігноруючи адреси, що вказані в сертифікаті підписувача. Опціональний, за замовчанням дорівнює false
nonceLen	Integer	Довжина одноразового випадкового числа в TSP-запиті. Діапазон значень: 0, 4..32. Якщо значення дорівнює 0 або знаходиться не в діапазоні, то випадкове число в TSP-запиті не використовується. Опціональний, за замовчанням дорівнює 8
policyId	OID	Ідентифікатор політики TSP-сервісу. Якщо поле відсутнє або в значенні пустий рядок, то параметр політики TSP-сервісу в запиті не використовується. Опціональний
url	String[] або String	масив URL-адрес TSP-сервісів. У випадку однієї адреси можна задати її, як рядок. Опціональний

Структура CMPROVIDERS_PARAMS

Назва поля	Тип	Опис
dir	String	Повний шлях до каталогу з бібліотеками провайдерів НКІ
allowedProviders	Object[] CMPROVIDER_PARAMS	Масив з параметрами провайдерів НКІ

Структура CMPROVIDER_PARAMS

Назва поля	Тип	Опис
lib	String	Назва файлу бібліотеки провайдера НКІ
config	Object	Параметри, специфічні для цього провайдера НКІ. Опціональний

Приклад файлу конфігурації:

```
{
  "cmProviders": {
    "allowedProviders": [ {
        "lib": "cm-diamond"
      }, {
        "lib": "cm-pkcs12",
        "config": {
          "createPfx": {
            "bagCipher": "2.16.840.1.101.3.4.1.22",
            "bagKdf": "1.2.840.113549.2.10",
            "iterations": 10000,
            "macAlgo": "2.16.840.1.101.3.4.2.2"
          }
        }
      }
    ]
  },
  "certCache": {
    "path": "C:/uapki/certs/",
    "trustedCerts": ["MIIE...a2s=", ... ]
  },
  "crlCache": {
    "path": "C:/uapki/certs/crls/"
  },
  "offline": false
}
```

Структура поля result у відповіді

Назва поля	Тип	Опис
certCache	Object CERT_CACHE_INFO	Інформація про стан кешу сертифікатів
crlCache	Object CRL_CACHE_INFO	Інформація про стан кешу СВС
countCmProviders	Integer	Кількість завантажених провайдерів НКІ
offline	Boolean	Режим роботи “офлайн”
ocsp	Object OCSP_INFO	Інформація про параметри OCSP-сервісу
proxy	Object PROXY_INFO	Інформація про параметри PROXY-сервісу
tsp	Object TSP_INFO	Інформація про параметри TSP-сервісу

Структура CERT_CACHE_INFO

Назва поля	Тип	Опис
countTrustedCerts	Integer	Кількість довірених сертифікатів у кеші сертифікатів
countCerts	Integer	Загальна кількість сертифікатів у кеші сертифікатів

Структура CRL_CACHE_INFO

Назва поля	Тип	Опис
countCrls	Integer	Кількість СВС у кеші СВС

Структура OCSP_INFO

Назва поля	Тип	Опис
nonceLen	Integer	Довжина одноразового випадкового nonce в OCSP-запиті. Якщо значення дорівнює 0, то випадковий nonce в OCSP-запиті не використовується

Структура PROXY_INFO

Назва поля	Тип	Опис
url	String	URL-адреса PROXY-сервісу. Якщо містить пустий рядок, то PROXY-сервіс не використовується.

Структура TSP_INFO

Назва поля	Тип	Опис
certReq	Boolean	Вимога повертати в TSP-відповіді сертифікат сервісу
forced	Boolean	Вимога використовувати URL-адреси TSP-сервісів, які задані в полі "url", ігноруючи адреси, що вказані в сертифікаті підписувача
nonceLen	Integer	Довжина одноразового випадкового числа в TSP-запиті. Якщо значення дорівнює 0, то випадкове число в запиті не використовується
policyId	OID	Ідентифікатор політики TSP-сервісу
url	String	URL-адреса TSP-сервісу. Масив URL-адрес повертається у вигляді рядка з роздільником ";"

Приклад запиту з вхідними параметрами, заданими безпосередньо:

```
{
  "method": "INIT",
  "parameters": {
    "cmProviders": {
      "dir": "C:/uapki/cm-libs/",
      "allowedProviders": [ {
          "lib": "cm-diamond"
        }, {
          "lib": "cm-pkcs12"
        }
      ]
    },
    "certCache": {
      "path": "C:/uapki/certs/",
      "trustedCerts": ["MIIE...a2s=", ... ]
    },
    "crlCache": {
      "path": "C:/uapki/certs/crls/"
    },
    "offline": false,
    "ocsp": {
      "nonceLen": 20
    },
    "tsp": {
      "certReq": true,
      "forced": true,
      "nonceLen": 8,
      "policyId": "1.2.804.2.1.1.1.2.3.1",
      "url": "http://url_ca/services/tsp/"
    }
  }
}
```

Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "INIT",
  "result": {
    "certCache": {
      "countCerts": 29,
      "countTrustedCerts": 5
    },
    "crlCache": {
      "countCrls": 4
    },
    "countCmProviders": 2,
    "offline": false,
    "ocsp": {
      "nonceLen": 20
    },
    "proxy": {
      "url": ""
    },
    "tsp": {
      "certReq": true,
      "forced": true,
      "nonceLen": 8,
      "policyId": "1.2.804.2.1.1.1.2.3.1",
      "url": "http://url_ca/services/tsp/"
    }
  }
}
```

Метод DEINIT

Метод призначено для звільнення ресурсів бібліотеки, які були виділені при ініціалізації.
Вхідні параметри: відсутні.
Вихідні параметри: відсутні.

Приклад запиту:

```
{
  "method": "DEINIT"
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "DEINIT",
  "result": {}
}
```

Метод PROVIDERS

Метод призначено для отримання переліку завантажених провайдерів НКІ та інформації про них.
Вхідні параметри: відсутні.

Структура поля result у відповіді

Назва поля	Тип	Опис
providers	Object[] PROVIDER_INFO[]	Масив інформації про завантажені провайдери НКІ

Структура PROVIDER_INFO

Назва поля	Тип	Опис
id	String	Ідентифікатор провайдера. Унікальне значення, наприклад: "PKCS12", "TOKEN"
apiVersion	String	Версія API провайдера в форматі major.minor.build
libVersion	String	Версія бібліотеки провайдера в форматі major.minor.build
description	String	Короткий опис провайдера
manufacturer	String	Назва виробника провайдера
supportListStorages	Boolean	Позначка підтримки методу "STORAGES"

Приклад запиту:

```
{
  "method": "PROVIDERS"
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "PROVIDERS",
  "result": {
    "providers": [{
      "id": "DIAMOND",
      "apiVersion": "1.0.0",
      "libVersion": "1.0.12",
      "description": "DIAMOND token library",
      "manufacturer": "SPECINFOSYSTEMS LLC",
      "supportListStorages": true
    }, {
      "id": "PKCS12",
      "apiVersion": "1.0.0",
      "libVersion": "1.0.8",
      "description": "PKCS12(PFX) file key storage library",
      "manufacturer": "SPECINFOSYSTEMS LLC",
      "supportListStorages": false
    }]
  }
}
```

Метод STORAGES

Метод призначено для отримання переліку НКІ визначеного провайдера.
Провайдери можуть не підтримувати даний метод, наприклад, провайдер PKCS12.

Структура поля parameters у запиті

Назва поля	Тип	Опис
provider	String	Ідентифікатор провайдера

Структура поля result у відповіді

Назва поля	Тип	Опис
storages	Object[] STORAGE_INFO[]	Масив інформації про наявні НКІ

Структура поля STORAGE_INFO

Назва поля	Тип	Опис
id	String	Ідентифікатор НКІ
description	String	Опис НКІ
manufacturer	String	Виробник НКІ
model	String	Модель НКІ
serial	String	Серійний номер НКІ
label	String	Текстове позначення НКІ
passwordCountLow	Boolean	Попередня спроба введення пароля була неправильною, кількість спроб введення паролю зменшена
passwordFinalTry	Boolean	Остання спроба вводу пароля
passwordLocked	Boolean	Пароль заблоковано
passwordToBeChanged	Boolean	Ознака, що потрібно змінити пароль
passwordAttemptsLeft	Integer	Кількість спроб, що залишилося до блокування пароля
passwordMinLen	Integer	Мінімальна довжина пароля
passwordMaxLen	Integer	Максимальна довжина пароля

Приклад запиту:

```
{
  "method": "STORAGES",
  "parameters": {
    "provider": "TOKEN"
  }
}
```

Приклад відповіді, якщо метод підтримується:

```
{
  "errorCode": 0,
  "method": "STORAGES",
  "result": {
    "storages": [ {
      "id": "1099999",
      "description": "DIAMOND token",
      "manufacturer": "SPECINFOSYSTEMS LLC",
      "model": "DIAMOND 1000",
      "serial": "1099999",
      "label": "",
      "passwordCountLow": false,
      "passwordFinalTry": false,
      "passwordLocked": false,
      "passwordToBeChanged": false,
      "passwordAttemptsLeft": 10,
      "passwordMinLen": 4,
      "passwordMaxLen": 64
    }, {
      ...
    } ]
  }
}
```

Приклад відповіді, якщо метод не підтримується:

```
{
  "errorCode": 4123,
  "method": "STORAGES",
  "result": {},
  "error": "UNSUPPORTED_CM_API"
}
```

Метод STORAGE_INFO

Метод призначено для отримання інформації про НКІ.
Провайдери можуть не підтримувати даний метод, наприклад, провайдер PKCS12.
Якщо провайдер підтримує даний метод, то в якості вихідних параметрів буде повернута структура STORAGE_INFO (див. метод STORAGES).

Структура поля parameters у запиті

Назва поля	Тип	Опис
provider	String	Ідентифікатор провайдера
storage	String	Ідентифікатор НКІ. Наприклад, це може бути ім'я файлу чи URL-адреса

Приклад запиту:

```
{
  "method": "STORAGE_INFO",
  "parameters": {
    "provider": "PKCS12",
    "storage": "storage-id"
  }
}
```

Приклад відповіді, якщо метод підтримується:

```
{
  "errorCode": 0,
  "method": "STORAGE_INFO",
  "result": {
    "id": "1099999",
    "description": "DIAMOND token",
    "manufacturer": "SPECINFOSYSTEMS LLC",
    "model": "DIAMOND 1000",
    "serial": "1099999",
    "label": "",
    "passwordCountLow": true,
    "passwordFinalTry": false,
    "passwordLocked": false,
    "passwordToBeChanged": false,
    "passwordAttemptsLeft": 9,
    "passwordMinLen": 4,
    "passwordMaxLen": 64
  }
}
```

Приклад відповіді, якщо метод не підтримується:

```
{
  "errorCode": 4123,
  "method": "STORAGE_INFO",
  "result": {},
  "error": "UNSUPPORTED_CM_API"
}
```

Метод OPEN

Метод призначено для авторизації користувача та відкриття НКІ.

Увага! Одночасно може бути відкритий тільки один НКІ, який доступний для всіх потоків.

Якщо при відкритті НКІ на ньому будуть знайдені сертифікати, вони будуть автоматично доступні для використання в інших методах бібліотеки.

Метод має три обов'язкових вхідних параметри ("provider", "storage" та "password"), один опціональний параметр "mode", а також може мати специфічні параметри, які залежать від конкретного провайдера НКІ.

Режими роботи з НКІ ("mode")

Значення	**Тип**
"RW"	Доступні всі методи для роботи з ключами. Режим за замовчанням
"RO"	Доступні методи для роботи з ключами, які не змінюють НКІ (аналог режиму "тільки читання" для звичайних файлів)
"CREATE"	Створення нового файлу PKCS#12, доступні всі методи для роботи з ключами (тільки для провайдера PKCS12)

PKCS12-провайдер має специфічний параметр — "openParams", який є опціональним. Він може використовуватися при створенні файлового контейнера або при роботі з ключами, які зберігаються не у файлі, а в оперативній пам’яті. Спеціальний режим роботи "в пам’яті" дозволяє використовувати вміст файлу у форматі PKCS12, що знаходиться в оперативній пам’яті, замість роботи з файлом на диску. Для використання режиму роботи "в пам’яті" необхідно в параметрі "storage" вказати ключове слово "file://memory" і задати зміст файлу в base64-кодировці в полі "bytes" параметру "openParams".

При створенні нових файлових контейнерів PKCS#12 в полі "createPfx" параметру "openParams" можна задати значення параметрів "bagCipher", "bagKdf", "macAlgo" та "iterations". Якщо поле "createPfx" не задано, то провайдер буде використовувати значення параметрів за замовчанням.

Параметри "createPfx" за замовчанням

Параметр	**Значення**	**Опис**
bagCipher	`"2.16.840.1.101.3.4.1.42"`	AES256-CBC-PAD
bagKdf	`"1.2.840.113549.2.11"`	HMAC-SHA-512
macAlgo	`"2.16.840.1.101.3.4.2.3"`	SHA-512
iterations	`10000`	

Структура поля parameters у запиту до PKCS12-провайдера

Назва поля	**Тип**	**Опис**
provider	String	Ідентифікатор PKCS12-провайдера
storage	String	Ім'я файлу
mode	String	Режим роботи з НКІ. Опціональний
password	String	Пароль
openParams	Object	Додаткові параметри

Структура поля result у відповіді

Назва поля	Тип	Опис
id	String	Ідентифікатор НКІ
description	String	Опис НКІ
manufacturer	String	Виробник НКІ
model	String	Модель НКІ
serial	String	Серійний номер НКІ
label	String	Текстовий опис НКІ, що задається користувачем
mechanisms	Object[]	Масив структур із описом механізмів, які доступні для використання

PKCS12-провайдер при вдалому відкритті файлу для параметрів "model", "serial" та "label" повертає пусті рядки.

Приклад запиту до PKCS12-провайдера для створення нового файлу ключів:

```
{
  "method": "OPEN",
  "parameters": {
    "provider": "PKCS12",
    "storage": "file.p12",
    "mode": "CREATE",
    "password": "password",
    "openParams": {
      "createPfx": {
        "bagCipher": "2.16.840.1.101.3.4.1.2",
        "bagKdf": "1.2.840.113549.2.9",
        "macAlgo": "2.16.840.1.101.3.4.2.1",
        "iterations": 10000
      }
    }
  }
}
```

Приклад запиту до PKCS12-провайдера для роботи із файлом ключів:

```
{
  "method": "OPEN",
  "parameters": {
    "provider": "PKCS12",
    "storage": "file.p12",
    "mode": "RW",
    "password": "password"
  }
}
```

Приклад запиту до PKCS12-провайдера для роботи з ключами "в пам’яті":

```
{
  "method": "OPEN",
  "parameters": {
    "provider": "PKCS12",
    "storage": "file://memory",
    "password": "password",
    "mode": "RO",
    "openParams": {
      "bytes": "MIIE4wIBAzCCBIAGCSqG...U+soAgInEA=="
    }
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "OPEN",
  "result": {
    "id": "file.p12",
    "description": "PKCS12-file session",
    "manufacturer": "2021 SPECINFOSYSTEMS LLC",
    "model": "",
    "serial": "",
    "label": "",
    "mechanisms": [
      {
        "id": "1.2.804.2.1.1.1.1.3.6",
        "name": "DSTU-4145",
        "keyParam": ["1.2.804.2.1.1.1.1.3.1.1.2.5", ... ],
        "signAlgo": ["1.2.804.2.1.1.1.1.3.6.1", ... ]
      }, {
        "id": "1.2.840.10045.2.1",
        "name": "ECDSA",
        "keyParam": ["1.2.840.10045.3.1.7", ... ],
        "signAlgo": ["1.2.840.10045.4.3.2", ... ]
      }, {
        ...
      }
    ]
  }
}
```

Метод CLOSE

Закриває поточний відкритий НКІ.
Вхідні параметри: відсутні.
Вихідні параметри: відсутні.

Приклад запиту:

```
{
  "method": "CLOSE"
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "CLOSE",
  "result": {}
}
```

Метод KEYS

Метод призначено для отримання переліку ключів на відкритому НКІ.
Вхідні параметри: відсутні.

Структура поля result у відповіді

Назва поля	Тип	Опис
keys	Object[] KEY_INFO[]	Масив інформації про ключі

Структура KEY_INFO

Назва поля	Тип	Опис
id	String	Ідентифікатор ключа. Унікальне значення
mechanismId	OID	Ідентифікатор алгоритму ключа
parameterId	OID або String	Ідентифікатор параметра ключа: OID – ідентифікатор EC-кривої; String – довжина RSA ключа в бітах (число)
signAlgo	OID[]	Масив ідентифікаторів алгоритму підпису, що підтримуються ключем
label	String	Текстове позначення ключа
application	String	Текстове позначення застосунка

Приклад запиту:

```
{
  "method": "KEYS"
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "KEYS",
  "result": {
    "keys": [
      {
        "id": "112233445566...DDEEFF00",
        "mechanismId": "1.2.804.2.1.1.1.1.3.1.1",
        "parameterId": "1.2.804.2.1.1.1.1.3.1.1.2.6",
        "signAlgo": ["1.2.804.2.1.1.1.1.3.1.1", ... ],
        "label": "DSTU-4145, M257_PB",
        "application": ""
      }, {
        "id": "CAFE8A8E1234...00000001",
        "mechanismId": "1.2.840.10045.2.1",
        "parameterId": "1.2.840.10045.3.1.7",
        "signAlgo": ["1.2.840.10045.4.3.2", ... ],
        "label": "ECDSA, prime256v1",
        "application": ""
      }
    ]
  }
}
```

Метод SELECT_KEY

Метод призначено для вибору поточного ключа у відкритому НКІ.

Увага! Одночасно може бути вибраний тільки один ключ, який доступний для всіх потоків. При виборі іншого ключа він зміниться для всіх потоків програми.

Структура поля parameters у запиті

Назва поля	Тип	Опис
id	String	Ідентифікатор ключа. Унікальний

Структура поля result у відповіді

Назва поля	Тип	Опис
id	String	Ідентифікатор ключа
mechanismId	OID	Ідентифікатор алгоритму ключа
parameterId	OID або String	Ідентифікатор параметра ключа: OID – ідентифікатор EC-кривої; String – довжина RSA ключа в бітах (число)
signAlgo	OID[]	Масив ідентифікаторів алгоритму підпису, що підтримуються ключем
label	String	Текстове позначення ключа
application	String	Текстове позначення застосунка
certId	Base64	Ідентифікатор сертифікату в кеші сертифікатів. Опціональний
certificate	Base64	Сертифікат ключа (за стандартом x.509) у форматі base64. Опціональний
exportable	Boolean	Можливість експортування ключа з НКІ. Опціональний
extAuth	String	Параметри додаткової автентифікації. Опціональний

Приклад запиту:

```
{
  "method": "SELECT_KEY",
  "parameters": {
    "id": "112233445566...DDEEFF00"
  }
}
```

Приклад відповіді у випадку, коли в кеші сертифікатів немає відповідного сертифікату:

```
{
  "errorCode": 0,
  "method": "SELECT_KEY",
  "result": {
    "id": "112233445566...DDEEFF00",
    "mechanismId": "1.2.804.2.1.1.1.1.3.1.1",
    "parameterId": "1.2.804.2.1.1.1.1.3.1.1.2.6",
    "signAlgo": ["1.2.804.2.1.1.1.1.3.1.1", ... ],
    "label": "DSTU-4145, M257_PB",
    "application": ""
  }
}
```

Приклад відповіді у випадку, коли в кеші сертифікатів є відповідний сертифікат:


```
{
  "errorCode": 0,
  "method": "SELECT_KEY",
  "result": {
    "id": "112233445566...DDEEFF00",
    "mechanismId": "1.2.804.2.1.1.1.1.3.1.1",
    "parameterId": "1.2.804.2.1.1.1.1.3.1.1.2.6",
    "signAlgo": ["1.2.804.2.1.1.1.1.3.1.1", ... ],
    "label": "DSTU-4145, M257_PB",
    "application": ""
    "certId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
    "certificate": "MIIErjCCBFagAwIBAgIUFXe...NcYCFp23iPeya2s="
  }
}
```

Метод CREATE_KEY

Метод призначено для створення нового ключа у відкритому НКІ. Параметри за якими може бути створений ключ визначаються при відкритті НКІ. Якщо метод виконано успішно, то новий ключ стає поточним вибраним ключем у НКІ.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
mechanismId	OID	Ідентифікатор алгоритму ключа
parameterId	OID або String	Ідентифікатор параметра ключа: OID – ідентифікатор EC-кривої; String – довжина RSA ключа в бітах (число) Опціональний
label	String	Текстове позначення ключа. Опціональний
application	String	Текстове позначення застосунку. Опціональний

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
id	Hex	Ідентифікатор ключа. Унікальний

Приклад запиту:

```
{
  "method": "CREATE_KEY",
  "parameters": {
    "mechanismId": "1.2.804.2.1.1.1.1.3.1.1",
    "parameterId": "1.2.804.2.1.1.1.1.3.1.1.2.6",
    "label": "new DSTU4145-key, M257_PB"
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "CREATE_KEY",
  "result": {
    "id": "112233445566...DDEEFF00"
  }
}
```

Метод DELETE_KEY

Метод призначено для знищення ключа у відкритому НКІ.
Вихідні параметри: відсутні.

Структура поля parameters у запиті

Назва поля	Тип	Опис
id	Hex	Ідентифікатор ключа. Унікальний

Приклад запиту:

```
{
  "method": "DELETE_KEY",
  "parameters": {
    "id": "112233445566...DDEEFF00"
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "DELETE_KEY",
  "result": {}
}
```

Метод GET_CSR

Метод призначено для отримання запиту на формування сертифікату для поточного вибраного ключа. Якщо алгоритм підпису не вказаний, то використовується перший алгоритм підпису із списку signAlgo для ключа.

Структура поля parameters у запиті

Назва поля	Тип	Опис
signAlgo	String	Алгоритм підпису. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
bytes	Base64	Запит на формування сертифікату (за стандартом x.509)

Приклад запиту:

```
{
  "method": "GET_CSR",
  "parameters": {
    "signAlgo": "1.2.804.2.1.1.1.1.3.1.1"
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "GET_CSR",
  "result": {
    "bytes": "MIIBJTCBzgIBADAAMIGIMGA...xV235n6GixwS"
  }
}
```

Метод CHANGE_PASSWORD

Метод призначено для зміни паролю доступу до відкритого НКІ.
Вихідні параметри: відсутні.

Структура поля parameters у запиті

Назва поля	Тип	Опис
newPassword	String	Новий пароль

Приклад запиту:

```
{
  "method": "CHANGE_PASSWORD",
  "parameters": {
    "newPassword": "newpass"
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "CHANGE_PASSWORD",
  "result": {}
}
```

Метод INIT_KEY_USAGE

Метод призначено для ініціалізації використання ключа. Параметри запиту і результату залежать від НКІ. Більшість НКІ не потребує виклику цього метода.

Приклад запиту:

```
{
  "method": "INIT_KEY_USAGE",
  "parameters": {
    ...
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "INIT_KEY_USAGE",
  "result": {}
}
```

Метод SIGN

Метод призначено для підпису даних.

В офлайновому стані бібліотека може зробити підпис у форматі CAdES-BES - перевірка статусу сертифікату власника буде виконуватися із застосуванням CBC, не використовуючи OCSP.

В опціях (поле "options") можна вказати додаткові параметри підпису.

Коли параметр "ignoreCertStatus" має значення true, то під час підпису не буде перевірятися статус сертифікату власника ключа. Опція доступна для використання лише для форматів підпису CAdES-BES та CAdES-T, для інших форматів підпису вона буде ігноруватися.

Опис форматів підпису наведено в Додатку Б. Назви формату підпису "CAdES-LT" та "CAdES-LTA" є синонімами "CAdES-XL" та "CAdES-A" відповідно.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
signParams	Object, SIGN_PARAMS	Набір параметрів підпису
dataTbs	Object[], DATA_TBS_PARAMS[]	Масив структур, що містять дані для підпису
options	Object, OPTION_PARAMS	Набір додаткових параметрів. Опціональний

Структура SIGN_PARAMS

Назва поля	**Тип**	**Опис**
signatureFormat	String	Формат підпису: "RAW", "CMS", "CAdES-BES", "CAdES-T", "CAdES-C", "CAdES-XL" ("CAdES-LT"), "CAdES-A" ("CAdES-LTA")
signAlgo	OID	Ідентифікатор алгоритму підпису. Опціональний
digestAlgo	OID	Ідентифікатор алгоритму гешування. Опціональний
detachedData	Boolean	Зовнішній підпис (дані не інкапсулюються). Опціональний, за замовченням true
includeCert	Boolean	Додати до підпису сертифікат власника ключа. Опціональний, за замовченням false
includeTime	Boolean	Додати до підпису час хосту (недовірений). Опціональний, за замовченням false
includeContentTS	Boolean	Додати до підпису позначку часу від даних. Опціональний, за замовченням false

Структура DATA_TBS_PARAMS

Назва поля	Тип	Опис
id	String	Ідентифікатор даних
bytes	Base64	Дані для підпису
file	String	Файл, який зберігає дані для підпису
ptr	Hex	Вказівник на пам’ять де зберігаються дані для підпису. Розмір вказівника залежіть від апаратно-програмної платформи
size	Integer	Кількість байт даних для підпису
type	OID	Ідентифікатор тип даних. Опціональний, за замовчанням має "1.2.840.113549.1.7.1" (дані)
isDigest	Boolean	Тип даних для підпису. Якщо true, то поле bytes містить геш, інакше оригінальні дані. Опціональний, за замовчанням false
signedAttributes	Object[], ATTRIBUTE_PARAMS[]	Масив структур, що містять дані атрибутів для підписаної частини підпису. Опціональний
unsignedAttributes	Object[], ATTRIBUTE_PARAMS[]	Масив структур, що містять дані атрибутів для непідписаної частини підпису. Опціональний

Структура ATTRIBUTE_PARAMS

Назва поля	Тип	Опис
type	OID	Ідентифікатор типу атрибуту
bytes	Base64	Дані атрибуту

Структура OPTION_PARAMS

Назва поля	Тип	Опис
ignoreCertStatus	Boolean	Не перевіряти статус сертифікату власника ключа. Опціональний, за замовчанням false

Структура поля result у відповіді

Назва поля	Тип	Опис
signatures	Object[], SIGNATURE_PARAMS[]	Масив структур SIGNATURE_PARAMS, що містять підписані дані

Структура SIGNATURE_PARAMS

Назва поля	Тип	Опис
id	String	Ідентифікатор даних
bytes	Base64	Дані підпису

Приклад запиту:


```
{
  "method": "SIGN",
  "parameters": {
    "signParams": {
      "signatureFormat": "CAdES-BES",
      "signAlgo": "1.2.804.2.1.1.1.1.3.1.1",
      "detachedData": true,
      "includeCert": false,
      "includeTime": true
    },
    "dataTbs": [
      {
        "id": "doc-0",
        "bytes": "VGhlIHF1aWNrIGJyb...p5IGRvZw=="
      }, {
        "id": "doc-1",
        "bytes": "VHJpcGxlIENyb3duIG9mIE1vdG9yc3BvcnQ="
      }
    ]
  }
}
```


Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "SIGN",
  "result": {
    "signatures": [
      {
        "id": "doc-0",
        "bytes": "MIISxQYJKoZIhvcNAQcCo...y26i8X+13kQ/l6"
      }, {
        "id": "doc-1",
        "bytes": "MIIHcgYJKoZIhvcNAQcCo...C719o6rNlQUrTOsBx8="
      }
    ]
  }
}
```

Метод VERIFY

Метод призначено для валідації підпису у CMs/CAdES-форматі або у форматі виходу криптографічних примітиві« ("»AW"). Опис форматів підпису наведено в Додатку Б. По«е "signat»re" є о’ов'язковим — в ньому підписані дані зберігаються в по«і "signature.by»es".

Для валдації CMs/CAdES-формату підпису, як зовнішнього підпису, додатково вказують оригінальні дані в полі "signature.content". Якщо CMS/CAdES-формату підпис має інкапсульовані дані, то поле "signature.content" не використовується.

Метод підтримує три типа валідації підпису CMS/CAdES-формату (поле "options.validationType"):
1) "STRUCT" — валідація структури підпису (за замовчанням), перевіряється структура даних підпису, електронний підпис підписаних атрибутів та позначок часу (за їх наявності);
2) "CHAIN" — валідація структури підпису та ланцюжка сертифікатів, включає в себе пункт 1 і створення ланцюжка сертифікатів підписувача та позначок часу (за їх наявності) за якими можна перевірити дійсність ланцюжка сертифікатів;
3) "FULL" — повна валідація підпису, включає до себе пункт 2 і валідацію дійсності всіх сертифікатів у ланцюжку на момент створення підпису.

Для спрощення аналізу результатів валідації підпису можна використовувати поля результату "validSignatures", "validDigests" та "bestSignatureTime". Поле "bestSignatureTime" містить найкращий довірений час підпису (в порядку пріоритету: "signingTime", "contentTS.genTime" та "signatureTS.genTime").

Коли використовується тип валідації "CHAIN" або "FULL" в полі "certificateChain" зберігається інформація про ланцюжок сертифікатів (масив записів CERT_CHAIN_INFO).
Якщо для валідації не вистачає сертифікату, то інформація для його пошуку зберігаються в полі "expectedCerts" (масив записів EXPECTED_CERT_INFO).
Якщо для визначення статусу сертифікату не вистачає СВС, то інформація для його пошуку зберігаються в полі "expectedCerts" (масив записів EXPECTED_CRL_INFO).

Поле "warnings" містить зауваження до результату валідації підпису (масив текстових рядків).
Під час повної валідації підпису для визначення статусу сертифікату використовується СВС. У випадку коли неможливо отримати СВС буде використовуватися OCSP-запит (онлайн). Щоб заборонити використання OCSP-сервісу необхідно встановити параметр "options.onlyCrl" в значення true (за замовчанням — false).

Для валідації RAW-формату підпису необхідно більше параметрів:
1) поле "signParams" містить параметри підпису ("signAlgo" - обов'язково);
2) поле "signerPubkey" містить параметри публічного ключа підписувача;
3) поле "signature.content" містить оригінальні дані.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
signature	Object SIGNATURE_DATA	Структура SIGNATURE_DATA, що містить підписані дані
signParams	Object SIGN_PARAMS	Набір параметрів підпису. Умовно-опціональний
signerPubkey	Object SIGNER_PUBKEY	Набір параметрів підписувача. Умовно-опціональний
options	Object OPTION_PARAMS	Набір додаткових параметрів. Опціональний

Структура SIGNATURE_DATA

Назва поля	Тип	Опис
bytes	Base64	Підписані дані
content	Base64	Оригінальні дані. Умовно-опціональний
file	String	Файл, який зберігає дані підпису
ptr	Hex	Вказівник на пам’ять де зберігаються дані підпису. Розмір вказівника залежіть від апаратно-програмної платформи
size	Integer	Кількість байт даних підпису
isDigest	Boolean	Якщо встановлений в true, то поле content містить геш від даних. За замовченням false

Структура SIGN_PARAMS

Назва поля	Тип	Опис
signAlgo	OID	Ідентифікатор алгоритму підпису

Структура SIGNER_PUBKEY

Назва поля	Тип	Опис
certificate	Base64	Сертифікат підписувача. Опціональний
certId	Base64	Ідентифікатор сертифікату підписувача. Опціональний
spki	Base64	Відкритий ключ підписувача з параметрами ключа, структура SubjectPublicKeyInfo за стандартом “x.509”. Опціональний

Структура OPTION_PARAMS

Назва поля	Тип	Опис
validationType	String	Тип валідації підпису: "STRUCT", "CHAIN", "FULL". Опціональний, за замовченням має "STRUCT"
verifySignerInfoIndex	Integer	Перевірити окремого користувача (перший індекс дорівнює 0). Якщо індекс дорівнює -1, то перевіряються всі користувачі. Опціональний, за замовченням -1
onlyCrl	Boolean	Використовувати виключно СВС. Опціональний, за замовченням false

В залежності від формату підписаних даних структура поля result відповіді відрізняються.

Структура поля result у відповіді валідації підписаних даних у “CMS/CAdES”-форматі

Назва поля	Тип	Опис
content	Object CONTENT_INFO	Структура CONTENT_INFO
certIds	Base64[]	Масив ідентифікаторів сертифікатів, які присутні в підписаних даних
signatureInfos	Object[] SIGNATURE_INFO[]	Масив інформації по кожному підпису

Структура поля result у відповіді валідації підписаних даних у “RAW”-форматі

Назва поля	Тип	Опис
statusSignature	String	Статус електронного підпису: "VALID", "INVALID", "FAILED"

Структура CONTENT_INFO

Назва поля	Тип	Опис
type	OID	Ідентифікатор типу даних
bytes	Base64	Інкапсульовані дані. Опціональний

Структура SIGNATURE_INFO

Назва поля	Тип	Опис
signerCertId	Base64	Ідентифікатор сертифікату підписувача. Опціональний
signatureFormat	String	Формат CMS/CAdES-підпису: "CMS", "CAdES-BES", "CAdES-T", "CAdES-C", "CAdES-XL", "CAdES-A"
status	String	Статус підписаних даних: "UNDEFINED", "INDETERMINATE", "TOTAL-FAILED", "TOTAL-VALID"
validSignatures	Boolean	Всі криптографічні підписи, що відносяться до структури формату підпису, валідні
validDigests	Boolean	Всі геши, що відносяться до структури формату підпису, валідні
bestSignatureTime	Time	Найкращий час підпису
signAlgo	OID	Ідентифікатор алгоритма підпису
statusSignature	String	Статус електронного підпису: "UNDEFINED", "INDETERMINATE", "FAILED", "INVALID", "VALID WITHOUT KEYUSAGE", "VALID"
digestAlgo	OID	Ідентифікатор алгоритма гешування даних
statusMessageDigest	String	Статус цифрового дайджесту даних: "UNDEFINED", "INDETERMINATE", "FAILED", "INVALID", "VALID"
signingTime	Time	Локальний час підпису. Опціональний — присутній,

		якщо підписані дані мають відповідний атрибут
signaturePolicy	Object SIGN_POLICY_INFO	Політика підпису. Опціональний — присутній, якщо підписані дані мають відповідний атрибут
statusEssCert	String	Статус ідентифікації сертифікату підписника (опціональний): "UNDEFINED", "NOT PRESENT", "INDETERMINATE", "FAILED", "INVALID", "VALID"
contentTS	Object TIMESTAMP_INFO	Позначка часу від даних. Опціональний — присутній, якщо підписані дані мають відповідний атрибут
signatureTS	Object TIMESTAMP_INFO	Позначка часу від підпису. Опціональний — присутній, якщо підписані дані мають відповідний атрибут
statusCertificateRefs	String	Статус посилань на всі сертифікати в атрибуті certificateRefs (присутній у форматі підпису "CAdES-C", "CAdES-XL" та "CAdES-A"): "UNDEFINED", "NOT PRESENT", "INDETERMINATE", "FAILED", "INVALID", "VALID"
certificateRefs	Object[] CERT_REF_INFO[]	Масив посилань на всі сертифікати в атрибуті certificateRefs. Опціональний — присутній, якщо підпис має відповідний атрибут
certValues	Base64[]	Масив ідентифікаторів сертифікатів, які присутні в атрибуті certValues (присутній у форматі підпису "CAdES-XL" та "CAdES-A"). Опціональний
revocationRefs	Object[] REVOCATION_REF_INFO[]	Масив посилань на всі елементи відклику в атрибуті revocationRefs (присутній у форматі підпису "CAdES-C", "CAdES-XL" та "CAdES-A"). Опціональний
archiveTS	Object TIMESTAMP_INFO	Архівна позначка часу. Опціональний — присутній, якщо підписані дані мають відповідний атрибут
signedAttributes	Object[] ATTRIBUTE_PARAMS[]	Масив атрибутів, що зберігаються у полі signedAttributes
unsignedAttributes	Object[] ATTRIBUTE_PARAMS[]	Масив атрибутів, що зберігаються у полі unsignedAttributes. Опціональний
certificateChain	Object[] CERT_CHAIN_INFO[]	Масив результатів валідації ланцюжків сертифікатів. Опціональний — присутній коли тип валідації підпису "CHAIN" або "FULL"
expectedCerts	Object[] EXPECTED_CERT_INFO[]	Масив інформації по сертифікату, який необхідний для валідації підпису. Опціональний
expectedCrls	Object[] EXPECTED_CRL_INFO[]	Масив інформації по файлу СВС, який необхідний для валідації підпису. Опціональний
warnings	String[]	Масив зауважень до результату перевірки. Опціональний

Структура поля SIGN_POLICY_INFO

Назва поля	**Тип**	**Опис**
sigPolicyId	OID	Ідентифікатор політики підпису

Структура TIMESTAMP_INFO

Назва поля	Тип	Опис
genTime	Time	Значення позначки часу
policyId	OID	Ідентифікатор політики TSP
hashAlgo	OID	Ідентифікатор алгоритму гешування
hashedMessage	Base64	Значення гешу
statusDigest	String	Статус значення позначки часу: "UNDEFINED", "NOT PRESENT", "INDETERMINATE", "FAILED", "INVALID", "VALID"
statusSignature	String	Статус перевірки підпису в позначці часу: "UNDEFINED", "NOT PRESENT", "INDETERMINATE", "FAILED", "INVALID", "VALID"
signerCertId	Base64	Ідентифікатор сертифікату підписувача позначки часу. Опціональний — присутній, якщо сертифікат знайдено в кеші сертифікатів

Структура поля CERT_REF_INFO

Назва поля	Тип	Опис
certHash	Object HASH_INFO	Інформація про геш сертифікату
issuer	Object RDNAME_INFO	Опис сертифікату. Перелік елементів опису сертифікату наведено в Додатку Ґ. Опціональний
serialNumber	Hex	Серійний номер сертифікату. Опціональний
status	String	Статус відповідності гешу сертифікату і сертифікату

Структура поля HASH_INFO

Назва поля	Тип	Опис
hashAlgo	OID	Ідентифікатор алгоритму гешування
hashAlgoParams	Base64	Параметри алгоритму гешування (DER-кодування ASN1). Опціональний
hashValue	Base64	Значення гешування

Структура поля REVOCATION_REF_INFO

Назва поля	Тип	Опис
crlIds	Object CRLID_INFO	Масив посилань на СВС. Опціональний
ocspIds	Object[] OCSPID_INFO	Масив посилань на OCSP-відповіді. Опціональний
otherRev	Object ATTRIBUTE_PARAMS	Альтернативна інформація про відкликання. Опціональний

Структура поля CRLID_INFO

Назва поля	Тип	Опис
crlHash	Object HASH_INFO	Інформація про геш СВС
crlIdentifier	Object CRLIDENTIFIER_INFO	Ідентифікатор СВС. Опціональний

Структура поля CRLIDENTIFIER_INFO

Назва поля	Тип	Опис
crlIssuer	Object RDNAME_INFO	Опис СВС. Перелік елементів опису СВС наведено в Додатку Ґ. Опціональний
crlIssuedTime	Time	Час видання СВС
crlNumber	Hex	Номер СВС. Опціональний

Структура поля OCSPID_INFO

Назва поля	Тип	Опис
ocspIdentifier	Object OCSPIDENTIFIER_INFO	Ідентифікатор OCSP-відповіді
ocspHash	Object HASH_INFO	Інформація про геш OCSP-відповіді. Опціональний

Структура поля OCSPIDENTIFIER_INFO

Назва поля	Тип	Опис
responderId	Object RDNAME_INFO або Hex	Опис сертифікату або ідентифікатор ключа OCSP-сервісу. Опціональний. Перелік елементів опису сертифікату наведено в Додатку Ґ
producedAt	Time	Час створення OCSP-відповіді

Структура CERT_CHAIN_INFO

Назва поля	Тип	Опис
subjectCertId	Base64	Ідентифікатор сертифікату
CN	String	Назва власника сертифікату (commonName)
entity	String	Призначення: "UNDEFINED", "SIGNER", "INTERMEDIATE", "CRL", "OCSP", "TSP", "CA", "ROOT"
source	String	Джерело: "UNDEFINED", "SIGNATURE", "STORE"
validity	Object CERT_VALIDITY	Період дії сертифікату
expired	Boolean	Ознака, що закінчився термін дії сертифікату
selfSigned	Boolean	Ознака, що сертифікат самопідписаний
trusted	Boolean	Ознака, що сертифікат довірений
issuerCertId	Base64	Ідентифікатор сертифікату видавця. Опціональний
statusSignature	String	Статус електронного підпису сертифікату:

		"UNDEFINED", "INDETERMINATE", "FAILED", "INVALID", "VALID WITHOUT KEYUSAGE", "VALID"
validateByCRL	Object VALIDATE_BY_CRL_INFO	Результат перевірки сертифікату користувача з використанням СВС. Опціональний
validateByOCSP	Object VALIDATE_BY_OCSP_INFO	Результат перевірки сертифікату користувача з використанням OCSP. Опціональний
statusValidation	String	Статус валідації сертифікату: "UNDEFINED", "NONE", "VALID", "INVALID", "EXPIRED"

Структура поля CERT_VALIDITY

Назва поля	**Тип**	**Опис**
notBefore	Time	Дата з якої сертифікат починає бути дійсним
notAfter	Time	Дата з якої сертифікат перестає бути дійсним

Структура VALIDATE_BY_CRL_INFO

Назва поля	**Тип**	**Опис**
crlId	Base64	Ідентифікатор СВС в кеші СВС
CN	String	Назва власника видавця СВС (commonName)
thisUpdate	Time	Час створення поточного СВС
nextUpdate	Time	Час створення наступного СВС
crlNumber	Hex	Порядковий номер випуску СВС
deltaCrlIndicator	Hex	Номер повного випуску СВС. Опціональний
issuerCertId	Base64	Ідентифікатор сертифікату видавця. Опціональний
statusSignature	String	Статус електронного підпису СВС: "UNDEFINED", "INDETERMINATE", "FAILED", "INVALID", "VALID WITHOUT KEYUSAGE", "VALID"
status	String	Статус сертифікату: "UNDEFINED", "GOOD", "REVOKED", "UNKNOWN"
revocationReason	String	Підстава відкликання. Можливі наступні значення: "UNSPECIFIED", "KEY_COMPROMISE", "CA_COMPROMISE", "AFFILIATION_CHANGED", "SUPERSEDED", "CESSATION_OF_OPERATION", "CERTIFICATE_HOLD", "REMOVE_FROM_CRL", "PRIVILEGE_WITHDRAWN", "AA_COMPROMISE". Опціональний
revocationTime	Time	Час відкликання. Опціональний

Структура VALIDATE_BY_OCSP_INFO

Назва поля	**Тип**	**Опис**
source	String	Джерело: "UNDEFINED", "SIGNATURE", "STORE"
responseStatus	String	Статус OCSP-відповіді: "UNDEFINED", "SUCCESSFUL", "MALFORMED_REQUEST", "INTERNAL_ERROR", "TRY_LATER", "SIG_REQUIRED", "UNAUTHORIZED"

producedAt	Time	Час створення OCSP-відповіді
statusSignature	String	Статус електронного підпису OCSP-відповіді: "UNDEFINED", "NOT PRESENT", "INDETERMINATE", "FAILED", "INVALID", "VALID"
signerCertId	Base64	Ідентифікатор сертифікату підписувача OCSP-відповіді. Опціональний
status	String	Статус сертифікату: "UNDEFINED", "GOOD", "REVOKED", "UNKNOWN"
thisUpdate	Time	Час створення поточного запису OCSP
nextUpdate	Time	Час створення наступного запису OCSP. Опціональний
revocationReason	String	Підстава відкликання. Можливі наступні значення: "UNSPECIFIED", "KEY_COMPROMISE", "CA_COMPROMISE", "AFFILIATION_CHANGED", "SUPERSEDED", "CESSATION_OF_OPERATION", "CERTIFICATE_HOLD", "REMOVE_FROM_CRL", "PRIVILEGE_WITHDRAWN", "AA_COMPROMISE". Опціональний
revocationTime	Time	Час відкликання. Опціональний

Структура EXPECTED_CERT_INFO

Назва поля	**Тип**	**Опис**
entity	String	Призначення: "UNDEFINED", "SIGNER", "INTERMEDIATE", "CRL", "OCSP", "TSP", "CA", "ROOT"
issuer	Object RDNAME_INFO	Опис сертифікату. Перелік елементів опису сертифікату наведено в Додатку Ґ. Опціональний
serialNumber	Hex	Серійний номер сертифікату. Опціональний
keyId	Hex	Ідентифікатор ключа. Опціональний
responderId	Object RDNAME_INFO або Hex	Опис сертифікату або ідентифікатор ключа OCSP-сервісу. Опціональний. Перелік елементів опису сертифікату наведено в Додатку Ґ

Структура EXPECTED_CRL_INFO

Назва поля	**Тип**	**Опис**
authorityKeyId	Hex	Ідентифікатор ключа видавця
issuer	Object RDNAME_INFO	Опис СВС. Перелік елементів опису СВС наведено в Додатку Ґ. Опціональний
url	String	URL зберігання СВС. Опціональний
full	Object CRL_FULL_INFO	Інформація про повний СВС. Опціональний

Структура поля CRL_FULL_INFO

Назва поля	Тип	Опис
thisUpdate	Time	Час створення поточного СВС
nextUpdate	Time	Час створення наступного СВС
crlNumber	Hex	Порядковий номер випуску СВС

Приклад запиту для валідації підписаних даних у “RAW”-форматі:

```
{
  "method": "VERIFY",
  "parameters": {
    "signature": {
      "bytes": "MEYCIQCpNEQQ...b0Icmdl+yPst",
      "content": "MWkwGAYJKoZI...AgUj+NmRJtw="
    },
    "signParams": {
      "signAlgo": "1.2.840.10045.4.3.2"
    },
    "signerPubkey": {
      "certId": "MIH+MIHlMQsw...NQAAAFwAAAA="
    }
  }
}
```

Приклад запиту для валідації підписаних даних у “CMS/CAdES”-форматі:

```
{
  "method": "VERIFY",
  "parameters": {
    "signature": {
      "bytes": "MIIHkQYJKoZI...ZNcGXe57GF5j"
    }
  }
}
```

Приклад відповіді валідації підписаних даних у "CMS/CAdES"-форматі:

```
{
  "errorCode": 0,
  "method": "VERIFY",
  "result": {
    "content": {
      "type": "1.2.840.113549.1.7.1",
      "bytes": "QWxpY2UgYW5k...ZV9hbmRfQm9i"
    },
    "certIds": [ "MGwwVDELMAkG...CwAAAD0AAAA=", ... ],
    "signatureInfos": [{
      "signerCertId": "MGwwVDELMAkG...CwAAAD0AAAA=",
      "signatureFormat": "CAdES-T",
      "status": "TOTAL-VALID",
      "validSignatures": true,
      "validDigests": true,
      "bestSignatureTime": "2021-07-08 12:32:41",
      "signAlgo": "1.2.804.2.1.1.1.1.3.1.1",
      "statusSignature": "VALID",
      "digestAlgo": "1.2.804.2.1.1.1.1.2.1",
      "statusMessageDigest": "VALID",
      "signingTime": "2021-07-08 12:32:39",
      "statusEssCert": "VALID",
```

```
      "contentTS": {
        "genTime": "2021-07-08 12:32:40",
        "policyId": "1.2.804.2.1.1.1.2.3.1",
        "hashAlgo": "1.2.804.2.1.1.1.1.2.1",
        "hashedMessage": "DxNVEwtKggoeT...le3BwYCYMrIzM=",
        "statusDigest": "VALID",
        "statusSignature": "VALID",
        "signerCertId": "MIIBMTCCARcx...EAAADkAAAA"
      },
      "signatureTS": {
        "genTime": "2021-07-08 12:32:41",
        "policyId": "1.2.804.2.1.1.1.2.3.1",
        "hashAlgo": "1.2.804.2.1.1.1.1.2.1",
        "hashedMessage": "cgdf4po1UowRj...4QmGX3iyPAMFg=",
        ...
      },
      "signedAttributes": [
      {
        "type": "1.2.840.113549.1.9.3",
        "bytes": "BgkqhkiG9w0BBwE="
      },
      {
        "type": "1.2.840.113549.1.9.5",
        "bytes": "Fw0yMzAyMTUxNTI1MDda"
      },
      {
        "type": "1.2.840.113549.1.9.4",
        "bytes": "BCAPE1UTC0qCC...7cHBgJgysjMw=="
      },
      ...
      ]
    }]
  }
}
```

Приклад відповіді валідації підписаних даних у "RAW"-форматі:

```
{
  "errorCode": 0,
  "method": "VERIFY",
  "result": {
    "statusSignature": "VALID"
  }
}
```

Метод ENCRYPT

Метод призначено для зашифрування даних для одного або декількох отримувачів.

Структура поля parameters у запиті

Назва поля	Тип	Опис
content	Object CONTENT_PARAMS	Структура з даними і параметрами шифрування
recipientInfos	Object[], RECIPINFO_PARAMS	Масив структур, що містить параметри отримувачів
unprotectedAttrs	Object[], ATTRIBUTE_PARAMS[]	Масив структур, що містить дані атрибутів для незашифрованої частини даних. Опціональний

Структура CONTENT_PARAMS

Назва поля	Тип	Опис
bytes	Base64	Дані для шифрування
encryptionAlgo	OID	Ідентифікатор алгоритму шифрування
type	OID	Ідентифікатор типу даних. За замовченням "1.2.840.113549.1.7.1" (pkcs7-data)

Структура RECIPINFO_PARAMS

Назва поля	Тип	Опис
certId	Base64	Ідентифікатор сертифікату отримувача
kdfAlgo	OID	Ідентифікатор алгоритму функції формування ключа
keyWrapAlgo	OID	Ідентифікатор алгоритму захисту ключа. Опціональний, залежить від kdfAlgo

Структура ATTRIBUTE_PARAMS

Назва поля	Тип	Опис
type	OID	Ідентифікатор типу атрибуту
bytes	Base64	Дані атрибуту

Рекомендовані схеми шифрування даних

№	encryptionAlgo	kdfAlgo	keyWrapAlgo
1	"1.2.804.2.1.1.1.1.3.3.2"	"1.2.804.2.1.1.1.1.3.7"	"1.2.804.2.1.1.1.1.1.3.11"
2	"1.2.804.2.1.1.1.1.3.3.2"	"1.2.804.2.1.1.1.1.3.8"	"1.2.804.2.1.1.1.1.1.3.11"
3	"1.2.804.2.1.1.1.1.1.1.3"	"1.2.804.2.1.1.1.1.3.4"	"1.2.804.2.1.1.1.1.1.1.5"
4	"1.2.804.2.1.1.1.1.1.1.3"	"1.2.804.2.1.1.1.1.3.5"	"1.2.804.2.1.1.1.1.1.1.5"

Структура поля result у відповіді

Назва поля	Тип	Опис
bytes	Base64	Зашифровані дані

Приклад запиту:

```
{
  "method": "ENCRYPT",
  "parameters": {
    "content": {
      "bytes": "VGhlIHF1aWNrIGJyb...p5IGRvZw==",
      "encryptionAlgo": "1.2.804.2.1.1.1.1.1.1.3"
    },
    "recipientInfos": [
      {
        "certId": "MIH6MIHhMRYw...HgYAdKV2AA==",
        "kdfAlgo": "1.2.804.2.1.1.1.1.3.4"
      }
    ]
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "ENCRYPT",
  "result": {
    "bytes": "MIIBSAYJKoZIhvcNAQcDo...srvSh3rZYugDU="
  }
}
```

Метод DECRYPT

Метод призначено для розшифрування даних.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	Зашифровані дані

Структура поля result у відповіді

Назва поля	Тип	Опис
content	Object CONTENT_INFO	Структура CONTENT_INFO, що містить розшифровані дані
originatorCertId	Base64	Ідентифікатор сертифікату відправника
unprotectedAttrs	Object[]	Масив атрибутів у структурі ATTRIBUTE_PARAMS, що зберігаються у полі unprotectedAttrs. Опціональний

Структура CONTENT_INFO

Назва поля	Тип	Опис
bytes	Base64	Розшифровані дані
type	OID	Ідентифікатор типу даних

Приклад запиту:

```
{
  "method": "DECRYPT",
  "parameters": {
    "bytes": "MIIBSAYJKoZIhvcNAQcDo...srvSh3rZYugDU="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "DECRYPT",
  "result": {
    "content": {
      "bytes": "VGhlIHF1aWNrIGJyb...p5IGRvZw==",
      "type": "1.2.840.113549.1.7.1"
    },
    "originatorCertId": "MIH6MIHhMRYw...HgYAdKV2AA=="
  }
}
```

Метод ADD_CERT

Метод призначено для додавання сертифікатів до локального кешу сертифікатів або до НКІ. Сертифікати можуть додаватися до кешу на постійній основі (зі збереженням на диску), або тимчасово (тільки на час поточної сесії до виконання DEINIT або перезапуску програми). Якщо постійний кеш сертифікатів не ініціалізовано (не вказано шлях до відповідного каталогу при ініціалізації бібліотеки), додавання можливе тільки тимчасове.

Сертифікати можуть додаватись двома способами: масивом сертифікатів або пакетом сертифікатів (p7b-файл). Поля bundle та certificates не можуть бути в запиті одночасно.

Якщо сертифікат, що додається до кешу сертифікатів, вже зберігається в кеші, то він не буде доданий - у відповіді буде повернутий ідентифікатор існуючого сертифікату (ознака isUnique буде мати значення false).

Якщо поле storage має значення true, то сертифікати будуть збережені на поточному відкритому НКІ.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
bundle	Base64	Пакет сертифікатів. Опціональний
certificates	Base64[]	Масив сертифікатів. Опціональний
permanent	Boolean	Зберегти сертифікат у кеші сертифікатів. Опціональний, за замовчанням дорівнює false
storage	Boolean	Додати сертифікати до НКІ. Опціональний, за замовчанням дорівнює false

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
added	Object[] CERT_ADDED[]	Масив інформації про додані сертифікати

Структура CERT_ADDED

Назва поля	**Тип**	**Опис**
certId	Base64	Ідентифікатор сертифікату в кеші сертифікатів
isUnique	Boolean	Ознака унікальності сертифікату

Приклад запиту:

```
{
  "method": "ADD_CERT",
  "parameters": {
    "certificates": [ "MIIErjCCBFag...Fp23iPeya2s=", ... ]
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "ADD_CERT",
  "result": {
    "added": [ {
        "certId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
        "isUnique": true
      },
      ...
    ]
  }
}
```

Метод CERT_INFO

Метод призначено для отримання інформації про сертифікат. Сертифікат повинен відповідати стандарту x.509 та мати версію 3.
Метод повертає масив розширень сертифікату в тому порядку в якому вони зберігаються в сертифікаті. Якщо розширення сертифікату відомо бібліотеці, то воно буде декодоване. Перелік розширень сертифікату, які можуть бути декодованими в CERT_INFO наведено в Додатку В. Перелік полів опису власника та видавця сертифікату, які можуть бути декодованими наведено в Додатку Ґ.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
bytes	Base64	Сертифікат. Взаємовиключний до поля certId
certId	Base64	Ідентифікатор сертифікату. Взаємовиключний до поля bytes

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
bytes	Base64	Сертифікат в DER-кодуванні ASN1
version	Integer	Версія сертифікату
serialNumber	Hex	Унікальний номер сертифікату в ЦСК
issuer	Object	Опис видавця сертифікату
validity	Object CERT_VALIDITY	Період дії сертифікату
subject	Object	Опис власника сертифікату
subjectPublicKeyInfo	Object SUBJECT_PUBLICKEY_INFO	Публічний ключ власника сертифікату
extensions	Object[], EXTENSION_INFO[]	Масив розширень, які має сертифікат
signatureInfo	Object SIGNATURE_INFO	Електронний підпис сертифікату
selfSigned	Boolean	Ознака, що сертифікат самопідписаний

Структура SUBJECT_PUBLICKEY_INFO

Назва поля	**Тип**	**Опис**
bytes	Base64	DER-кодоване поле subjectPublicKeyInfo
algorithm	OID	Ідентифікатор алгоритму публічного ключа
parameters	Base64	Параметри алгоритму публічного ключа
publicKey	Base64	Значення публічного ключа

Структура EXTENSION_INFO

Назва поля	**Тип**	**Опис**
extnId	String	Ідентифікатор розширення
critical	Boolean	Ознака, що розширення критичне. Опціональний
extnValue	Base64	Закодоване значення розширення
decoded	Object DECODED_EXTENSION_INFO	Декодоване значення розширення. Опціональний

Структура DECODED_EXTENSION_INFO

Назва поля	Тип	Опис
id	String	Найменування розширення
value	Object	Значення розширення.

Структура SIGNATURE_INFO

Назва поля	Тип	Опис
algorithm	OID	Ідентифікатор алгоритму підпису
parameters	Base64	Параметри алгоритму підпису. Опціональний
signature	Base64	Значення підпису

Приклад запиту:

```
{
  "method": "CERT_INFO",
  "parameters": {
    "bytes": "MIIErjCCBFagAwIBAgIUFXeRu...NcYCFp23iPeya2s="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "CERT_INFO",
  "result": {
    "bytes": "MIIErjCCBFagAwIBAg...cYCFp23iPeya2s=",
    "version": 3,
    "serialNumber": "157791B9508857ED0400...0000",
    "issuer": {
      "C": "UA",
      "SERIALNUMBER": "UA-12345678-0001",
      "CN": "Центр сертифікації ключів",
      "O": "Test CA",
      "L": "Київ"
    },
    "validity": {
      "notBefore": "2020-08-26 12:34:56",
      "notAfter": "2022-08-26 12:34:56"
    },
    "subject": {
      "C": "UA",
      "CN": "Серпень Аугусто",
      "L": "Київ",
      "SN": "Серпень",
      "G": "Аугусто"
    },
    "subjectPublicKeyInfo": {
      "bytes": "MFkwEwYHKoZIzj0CAQY...nZOCZhbZMl3XsA==",
      "algorithm": "1.2.804.2.1.1.1.1.3.1.1",
      "parameters": "MFEGDSqGJAIBAQEBAwEB...uPrFeQQ=",
      "publicKey": "BCEhu7U+dG5kWwuTfPV30tf...8SjmlDitQE="
    },
    "extensions": [
      {
        "extnId": "2.5.29.14",
```

```
        "extnValue": "BCAzM/MjlbJMdildTG...98Wazw8wPoj+g==",
        "decoded": {
          "id": "subjectKeyIdentifier",
          "value": {
            "keyIdentifier": "BCB3BE7274D075DD...1370"
          }
        }
      },
      {
        "extnId": "2.5.29.35",
        "extnValue": "BCC8s75ydNB13VIlK2...PPVx/adALwTcA==",
        "decoded": {
          "id": "authorityKeyIdentifier",
          "value": {
            "keyIdentifier": "D0069AA0A8DF7D70...11A6"
          }
        }
      },
      {
        "extnId": "2.5.29.15",
        "critical": true,
        "extnValue": "AwIGwA==",
        "decoded": {
          "id": "keyUsage",
          "value": {
            "digitalSignature": true,
            "contentCommitment": true
          }
        }
      },
      ...
    ],
    "signatureInfo": {
      "algorithm": "1.2.804.2.1.1.1.1.3.1.1",
      "signature": "MIIErjCCBFagAwIBAg...cYCFp23iPeya2s="
    },
    "selfSigned": false
  }
}
```

Метод GET_CERT

Метод призначено для отримання сертифікату (в DER-кодуванні) із кешу сертифікатів. Метод повертає сертифікат, якщо він є в кеші сертифікатів, інакше повертається помилка - сертифікат не знайден ("CERT_NOT_FOUND").

Структура поля parameters у запиті

Назва поля	Тип	Опис
certId	Base64	Ідентифікатор сертифікату

Структура поля result у відповіді

Назва поля	Тип	Опис
bytes	Base64	Сертифікат

Приклад запиту:

```
{
  "method": "GET_CERT",
  "parameters": {
    "certId": "MIH+MIHlMQsw...NQAAAFwAAAA="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "GET_CERT",
  "result": {
    "bytes": "MIIErjCCBFag...Fp23iPeya2s="
  }
}
```

Метод LIST_CERTS

Метод призначено для отримання переліку ідентифікаторів сертифікатів із кешу сертифікатів.

Структура поля parameters у запиті

Назва поля	Тип	Опис
showCertInfos	Boolean	Вивести інформацію по кожному сертифікату. Опціональний, за замовчанням false
storage	Boolean	Позначка, що сертифікати зберігаються на НКІ. Опціональний, за замовчанням false
offset	Integer	Індекс першого сертифікату. Опціональний, за замовчанням 0
pageSize	Integer	Максимальна кількість сертифікатів. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
certIds	Base64[]	Масив ідентифікаторів сертифікатів
certInfos	Object[] CERT_INFO	Масив інформації по сертифікатах (структура CERT_INFO). Опціональний
count	Integer	Кількість сертифікатів
offset	Integer	Індекс першого сертифікату
pageSize	Integer	Максимальна кількість сертифікатів

Структура поля CERT_INFO

Назва поля	Тип	Опис
certId	Base64	Ідентифікатор сертифікату
serialNumber	Hex	Унікальний номер сертифікату в ЦСК
issuer	Object	Опис видавця сертифікату
validity	Object CERT_VALIDITY	Період дії сертифікату
subject	Object	Опис власника сертифікату
keyAlgo	OID	Ідентифікатор алгоритму ключа
subjectKeyIdentifier	Hex	Ідентифікатор ключа власника сертифікату
authorityKeyIdentifier	Hex	Ідентифікатор ключа видавця сертифікату
keyUsage	Object KEY_USAGE	Призначення ключа, опис структури KEY_USAGE дано в Додатку В
extKeyUsage	OID[]	Масив ідентифікаторів розширенного призначення ключа
isCa	Boolean	Ознака, що сертифікат ЦСК. Опціональний, за замовчанням false
isCmp	Boolean	Ознака, що сертифікат CMP-сервісу. Опціональний, за замовчанням false
isOcsp	Boolean	Ознака, що сертифікат OCSP-сервісу. Опціональний, за замовчанням false

isTsp	Boolean	Ознака, що сертифікат TSP-сервісу. Опціональний, за замовчанням false

Приклад запиту:

```
{
  "method": "LIST_CERTS",
  "parameters": {
    "offset": 10,
    "pageSize": 10
  }
}
```

Приклад відповіді (з інформацією по сертифікатах):

```
{
  "errorCode": 0,
  "method": "LIST_CERTS",
  "result": {
    "certIds": [ "MIGWMH4xCzAJ...QAAAFwAAAA=", ... ],
    "certInfos": [
      ..., {
        "certId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
        "serialNumber": "05E19E2CD92EA29...10000004A010000",
        "issuer": {
          "O": "Test CA",
          "CN": "Test CA",
          "C": "UA",
          "L": "Київ"
        },
        "validity": {
          "notBefore": "2023-05-08 08:30:00",
          "notAfter": "2028-05-08 08:30:00"
        },
        "subject": {
          "O": "Test CA",
          "CN": "OCSP-service",
          "C": "UA",
          "L": "Київ"
        },
        "keyAlgo": "1.2.804.2.1.1.1.1.3.1.1",
        "subjectKeyIdentifier": "BCB3BE7274D075DD...1370",
        "authorityKeyIdentifier": "D0069AA0A8DF7D70...11A6",
        "keyUsage": {
          "digitalSignature": true
        },
        "extKeyUsage": ["1.3.6.1.5.5.7.3.9"],
        "isOcsp": true
      },
      ...
    ],
    "count": 29,
    "offset": 10,
    "pageSize": 10
  }
}
```

Приклад відповіді (без інформації по сертифікатах):


```
{
  "errorCode": 0,
  "method": "LIST_CERTS",
  "result": {
    "certIds": [ "MIGWMH4xCzAJ...QAAAFwAAAA=", ... ],
    "count": 29,
    "offset": 10,
    "pageSize": 10
  }
}
```

Метод REMOVE_CERT

Метод видаляє сертифікат із кешу сертифікатів або з відкритого НКІ.

Якщо не вказано сертифікат (bytes та certId відсутні), який потрібно видалити, з кешу видаляються всі тимчасові сертифікати (які були додані методом ADD_CERT без ознаки *permanent = true* або додані опосередковано в інших методах).

Вихідні параметри: відсутні.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
bytes	Base64	Сертифікат. Опціональний
certId	Base64	Ідентифікатор сертифікату. Опціональний
permanent	Boolean	Видалити сертифікат із постійного кешу. Опціональний, за замовчанням дорівнює false
storage	Boolean	Видалити сертифікат з НКІ. Опціональний, за замовчанням дорівнює false

Приклад запиту:

```
{
  "method": "REMOVE_CERT",
  "parameters": {
    "certId": "MIH+MIHlMQsw...NQAAAFwAAAA="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "REMOVE_CERT",
  "result": {}
}
```

Метод VERIFY_CERT

Метод призначено для валідації сертифікату. Якщо сертифікат самопідписаний, то поле issuerCertId у відповіді відсутнє.

Поле validateTime встановлює значення часу, на який необхідно визначити валідність сертифікату. У разі наявності цього поля валідація виконується тільки за СВС. Якщо це поле відсутнє, то використовується поточний час.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	Сертифікат. Взаємовиключний до поля certId
certId	Base64	Ідентифікатор ключа. Взаємовиключний до поля certificate
validationType	String	Типи валідації сертифікату за статусом відкликання. Має наступні значення: "CRL" та "OCSP". Опціональний
validateTime	Time	Значення часу валідації. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
validateTime	Time	Значення часу перевірки валідації
subjectCertId	Base64	Ідентифікатор сертифікату користувача
validity	Object CERT_VALIDITY	Період дії сертифікату користувача
expired	Boolean	Ознака, що закінчився термін дії сертифікату
selfSigned	Boolean	Ознака, що сертифікат самопідписаний
trusted	Boolean	Ознака, що сертифікат довірений
statusSignature	String	Статус електронного підпису сертифікату: "VALID", "INVALID", "FAILED"
issuerCertId	Base64	Ідентифікатор сертифікату видавця. Опціональний
validateByCRL	Object VALIDATE_BY_CRL	Результат перевірки сертифікату користувача з використанням СВС. Опціональний
validateByOCSP	Object VALIDATE_BY_OCSP	Результат перевірки сертифікату користувача з використанням OCSP. Опціональний

Структура поля VALIDATE_BY_CRL

Назва поля	Тип	Опис
status	String	Статус сертифікату: "UNDEFINED", "GOOD", "REVOKED", "UNKNOWN"
revocationReason	String	Підстава відкликання. Можливі наступні значення: "UNDEFINED", "UNSPECIFIED", "KEY_COMPROMISE", "CA_COMPROMISE", "AFFILIATION_CHANGED", "SUPERSEDED", "CESSATION_OF_OPERATION", "CERTIFICATE_HOLD", "REMOVE_FROM_CRL", "PRIVILEGE_WITHDRAWN", "AA_COMPROMISE". Опціональний
revocationTime	Time	Час відкликання. Опціональний
full	Object CRL_INFO	Інформація про повний СВС
delta	Object CRL_INFO	Інформація про частковий СВС. Опціональний

Структура поля CRL_INFO

Назва поля	Тип	Опис
url	String	URL зберігання СВС. Опціональний
crlId	Base64	Ідентифікатор СВС в кеші СВС
statusSignature	String	Статус електронного підпису СВС: "VALID", "INVALID", "FAILED"

Структура поля VALIDATE_BY_OCSP

Назва поля	Тип	Опис
status	String	Статус сертифікату: "UNDEFINED", "GOOD", "REVOKED", "UNKNOWN"
revocationReason	String	Підстава відкликання. Можливі наступні значення: "UNDEFINED", "UNSPECIFIED", "KEY_COMPROMISE", "CA_COMPROMISE", "AFFILIATION_CHANGED", "SUPERSEDED", "CESSATION_OF_OPERATION", "CERTIFICATE_HOLD", "REMOVE_FROM_CRL", "PRIVILEGE_WITHDRAWN", "AA_COMPROMISE". Опціональний
revocationTime	Time	Час відкликання. Опціональний
responseStatus	String	Статус OCSP-відповіді: "UNDEFINED", "SUCCESSFUL", "MALFORMED_REQUEST", "INTERNAL_ERROR", "TRY_LATER", "SIG_REQUIRED", "UNAUTHORIZED"
responderId	Object RDNAME_INFO або Hex	Опис сертифікату або ідентифікатор ключа OCSP-сервісу. Опціональний. Перелік елементів опису сертифікату наведено в Додатку Ґ
statusSignature	String	Статус електронного підпису OCSP-відповіді: "VALID", "INVALID", "FAILED"
producedAt	Time	Час створення OCSP-відповіді
thisUpdate	Time	Час створення поточного запису OCSP
nextUpdate	Time	Час створення наступного запису OCSP. Опціональний

Приклад запиту:

```
{
  "method": "VERIFY_CERT",
  "parameters": {
    "bytes": "MIIErjCCBFag...Fp23iPeya2s="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "VERIFY_CERT",
  "result": {
    "validateTime": "2021-04-29 12:34:56",
    "subjectCertId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
    "validity": {
      "notBefore": "2020-08-26 23:13:07",
      "notAfter": "2022-08-26 23:13:07"
    },
    "expired": false,
    "selfSigned": false,
    "trusted": false,
    "statusSignature": "VALID",
    "issuerCertId": "MIH+MIHlMQsw...AQAAAAEAAAA="
  }
```

```
}
```

Приклад відповіді з OCSP-відповіддю:


```
{
  "errorCode": 0,
  "method": "VERIFY_CERT",
  "result": {
    "validateTime": "2021-04-29 12:34:56",
    "subjectCertId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
    "validity": {
      "notBefore": "2020-08-26 23:13:07",
      "notAfter": "2022-08-26 23:13:07"
    },
    "expired": false,
    "selfSigned": false,
    "trusted": false,
    "statusSignature": "VALID",
    "issuerCertId": "MIH+MIHlMQsw...AQAAAAEAAAA=",
    "validateByOCSP": {
      "status": "GOOD",
      "responseStatus": "SUCCESSFUL",
      "responderId": {
        "O": "Test CA",
        "CN": "OCSP-service",
        "C": "UA",
        "L": "Київ"
      },
      "statusSignature": "VALID",
      "producedAt": "2021-04-29 12:34:56",
      "thisUpdate": "2021-04-29 12:34:56"
    }
  }
}
```

Метод CERT_STATUS_BY_OCSP

Метод призначено для формування OCSP-запиту та отримання OCSP-відповіді. Метод дозволяє сформувати OCSP-запит по сертифікату видавника та серійному номеру сертифікату або за окремими параметрами. Якщо в запиті метода невказано поле url, то OCSP-запит не буде передаватися до OCSP-сервісу — результатом роботи метода буде лише поле requestBytes у відповіді метода.

Структура поля parameters у запиті

Назва поля	Тип	Опис
url	String	URL-адреса OCSP-сервісу. Опціональний
hashAlgo	OID	Ідентифікатор алгоритму гешування
issuerCertId	Base64	Ідентифікатор сертифікату видавця. Опціональний
serialNumber	Hex	Унікальний номер сертифікату в ЦСК
issuerBytes	Base64	Значення поля issuer видавця сертифікату. Опціональний
issuerNameHash	Hex	Значення гешу поля issuer видавця сертифікату. Опціональний
issuerKeyHash	Hex	Значення гешу відкритого ключа видавця сертифікату. Опціональний
nonceLen	Integer	Довжина одноразового випадкового числа в OCSP-запиті. Якщо значення дорівнює 0 або відсутнє, то випадкове число в OCSP-запиті не використовується. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
requestBytes	Base64	OCSP-запит
bytes	Base64	OCSP-відповідь
responseStatus	String	Статус OCSP-відповіді: "UNDEFINED", "SUCCESSFUL", "MALFORMED_REQUEST", "INTERNAL_ERROR", "TRY_LATER", "SIG_REQUIRED", "UNAUTHORIZED"
status	String	Статус сертифікату: "UNDEFINED", "GOOD", "REVOKED", "UNKNOWN"
certIds	Base64[]	Масив ідентифікаторів сертифікатів, які присутні в OCSP-відповіді
certId	Base64	Ідентифікатор сертифікату
producedAt	Time	Час створення OCSP-відповіді
thisUpdate	Time	Час створення поточного запису OCSP
nextUpdate	Time	Час створення наступного запису OCSP. Опціональний
responderId	Object RDNAME_INFO або Hex	Опис сертифікату або ідентифікатор ключа OCSP-сервіса. Опціональний. Перелік елементів опису сертифікату наведено в Додатку Ґ
statusSignature	String	Статус електронного підпису OCSP-відповіді: "VALID", "INVALID", "FAILED"

Приклад запиту:


```
{
  "method": "CERT_STATUS_BY_OCSP",
  "parameters": {
    "url": "http://url_ca/services/ocsp/",
    "hashAlgo": "1.2.804.2.1.1.1.1.2.1",
    "issuerCertId": "MIH+MIHlMQsw...AQAAAAEAAAA="
    "serialNumber": "157791B9508857ED04000000...0000"
  }
}
```


Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "CERT_STATUS_BY_OCSP",
  "result": {
    "requestBytes": "MHAwbjBsMGow...AKkeBgBzpXYA",
    "bytes": "MII2ZwoBAKCC...uvw4P7wsFTk=",
    "responseStatus": "SUCCESSFUL",
    "status": "GOOD",
    "certIds": [
      "MIH+MIHlMQsw...NQAAAFwAAAA=",
      "MIH+MIHlMQsw...AQAAAAEAAAA="
    ],
    "certId": "MIH+MIHlMQsw...NQAAAFwAAAA=",
    "producedAt": "2023-08-01 12:34:56",
    "thisUpdate": "2023-08-01 12:34:56",
    "responderId": {
      "O": "Test CA",
      "CN": "OCSP-service",
      "C": "UA",
      "L": "Київ"
    },
    "statusSignature": "VALID"
  }
}
```

Метод ADD_CRL

Метод призначено для додавання СВС (списку відкликаних сертифікатів) до кешу СВС. Повертає ідентифікатор СВС у кеші СВС. СВС можуть додаватися до кешу СВС на постійній основі (зі збереженням на диску), або тимчасово (тільки на час поточної сесії до виконання DEINIT або перезапуску програми). Якщо постійний кеш СВС не ініціалізовано (не вказано шлях до відповідного каталогу при ініціалізації бібліотеки), додавання СВС можливе тільки тимчасове.

СВС має відповідати стандарту x.509.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
bytes	Base64	Бінарні дані СВС
permanent	Boolean	Ознака зберегти СВС у кеші СВС

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
crlId	Base64	Ідентифікатор СВС
isUnique	Boolean	Ознака унікальності сертифікату

Приклад запиту:

```
{
  "method": "ADD_CRL",
  "parameters": {
    "bytes": "MIIJajCCCRIC...15Wd5gBHHCg="
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "ADD_CRL",
  "result": {
    "crlId": "MIHtMIHlMQsw...0ZfQsgIDAULT",
    "isUnique": true
  }
}
```

Метод CRL_INFO

Метод призначено для отримання інформації, яка зберігається в СВС.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	Бінарні дані СВС. Взаємовиключний до поля crlId
crlId	Base64	Ідентифікатор СВС. Взаємовиключний до поля bytes
showRevokedCerts	Boolean	Відображити список відкликаних сертифікатів. За замовчанням дорівнює true

Структура поля result у відповіді

Назва поля	Тип	Опис
issuer	Object	Опис видавця
thisUpdate	Time	Час створення поточного СВС
nextUpdate	Time	Час створення наступного СВС
countRevokedCerts	Integer	Кількість відкликанних сертифікатів
authorityKeyId	Hex	Ідентифікатор ключа CA
crlNumber	Hex	Порядковий номер випуску СВС
deltaCrlIndicator	Hex	Номер повного випуску СВС. Опціональний
revokedCerts	Object[] REVOKED_CERT_INFO[]	Масив відкликанних сертифікатів (структура REVOKED_CERT_INFO). Опціональний

Структура поля REVOKED_CERT_INFO

Назва поля	Тип	Опис
userCertificate	Hex	Серійний номер відкликаного сертифікату
revocationDate	Time	Час відкликання
crlReason	String	Підстава відкликання. Опціональний. Можливі наступні значення: "UNDEFINED", "UNSPECIFIED", "KEY_COMPROMISE", "CA_COMPROMISE", "AFFILIATION_CHANGED", "SUPERSEDED", "CESSATION_OF_OPERATION", "CERTIFICATE_HOLD", "REMOVE_FROM_CRL", "PRIVILEGE_WITHDRAWN", "AA_COMPROMISE"
invalidityDate	Time	Час недійсності

Приклад запиту:

```
{
  "method": "CRL_INFO",
  "parameters": {
    "bytes": "MIIJajCCCRIC...15Wd5gBHHCg="
  }
}
```

Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "CRL_INFO",
  "result": {
    "issuer": {
      "C": "UA",
      "SERIALNUMBER": "UA-12345678-0001",
      "CN": "Центр сертифікації ключів",
      "O": "Організація",
      "OU": "ЦСК",
      "L": "Київ"
    },
    "thisUpdate": "2021-07-31 06:00:00",
    "nextUpdate": "2021-08-07 06:00:00",
    "countRevokedCerts": 25,
    "authorityKeyId": "D0069AA0...9EA28CC7",
    "crlNumber": "0142D3",
    "revokedCerts": [
      {
        "userCertificate": "157791B95088...14000000",
        "revocationDate": "2019-04-17 15:16:22",
        "crlReason": "SUPERSEDED",
        "invalidityDate": "2019-04-17 15:16:22"
      }, {
        "userCertificate": "157791B95088...35000000",
        "revocationDate": "2019-11-07 14:20:06",
        "crlReason": "CERTIFICATE_HOLD",
        "invalidityDate": "2019-11-07 14:20:06"
      },
      ...
    ]
  }
}
```

Метод LIST_CRLS

Метод призначено для отримання переліку СВС в кеші СВС.

Структура поля parameters у запиті

Назва поля	Тип	Опис
showCrlInfos	Boolean	Вивести інформацію по кожному СВС. Опціональний, за замовчанням false
offset	Integer	Індекс першого СВС. Опціональний, за замовчанням 0
pageSize	Integer	Максимальна кількість СВС. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
crlIds	Base64[]	Масив ідентифікаторів СВС
crlInfos	Object[] CRL_INFO	Масив інформації по СВС (структура CRL_INFO). Опціональний
count	Integer	Кількість СВС
offset	Integer	Індекс першого СВС
pageSize	Integer	Максимальна кількість СВС

Структура поля CRL_INFO

Назва поля	Тип	Опис
crlId	Base64	Ідентифікатор СВС
issuer	Object	Опис видавця
thisUpdate	Time	Час створення поточного СВС
nextUpdate	Time	Час створення наступного СВС
countRevokedCerts	Integer	Кількість відкликанних сертифікатів
authorityKeyId	Hex	Ідентифікатор ключа CA
crlNumber	Hex	Порядковий номер випуску СВС
deltaCrlIndicator	Hex	Номер повного випуску СВС. Опціональний
isObsolete	Boolean	Ознака, що СВС застарів

Приклад запиту:

```
{
  "method": "LIST_CRLS",
  "parameters": {
    "offset": 0,
    "showCrlInfos": true
  }
}
```

Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "LIST_CRLS",
  "result": {
    "crlIds": [ "MIHpMIHhMRYw...RltGE0ZbQutC", ... ],
    "crlInfos": [ {
      "crlId": "MIHpMIHhMRYw...RltGE0ZbQutC",
      "issuer": {
        "C": "UA",
        "SERIALNUMBER": "UA-12345678-0001",
        "CN": "Центр сертифікації ключів",
        "O": "Організація",
        "OU": "ЦСК",
        "L": "Київ"
      },
      "thisUpdate": "2023-08-01 14:51:10",
      "nextUpdate": "2023-08-01 16:53:12",
      "countRevokedCerts": 118,
      "authorityKeyId": "D0069AA0...9EA28CC7",
      "crlNumber": "095178",
      "deltaCrlIndicator": "0947B1",
      "isObsolete": true
    }, {
      "crlId": "MIHpMIHhMRYw...MDMzAgMJj9g=",
      "issuer": {
        "C": "UA",
        "SERIALNUMBER": "UA-12345678-0001",
        "CN": "Центр сертифікації ключів",
        "O": "Організація",
        "OU": "ЦСК",
        "L": "Київ"
      },
      "thisUpdate": "2023-08-01 16:53:12",
      "nextUpdate": "2023-08-01 18:53:12",
      "countRevokedCerts": 123,
      "authorityKeyId": "D0069AA0...9EA28CC7",
      "crlNumber": "095179",
      "deltaCrlIndicator": "0947B1"
    },
    ...
    ],
    "count": 4,
    "offset": 0,
    "pageSize": 4
  }
}
```

Метод REMOVE_CRL

Метод призначено для видалення визначеного СВС (опціонально) та застарілих СВС з кешу СВС. Якщо в запиті вказано ідентифікатор СВС, то метод спочатку видаляє вказаний СВС, а потім всі застарілі СВС. Параметри відповіді – відсутні.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
crlId	Base64	Ідентифікатор СВС у СВС-кеші
permanent	Boolean	Ознака, щоб видалити СВС із місця зберігання СВС. Опціональний, за замовчанням дорівнює false

Приклад запиту:

```
{
  "method": "REMOVE_CRL",
  "parameters": {
    "crlId": "MIHtMIHlMQsw...0ZfQsgIDAULT",
    "permanent": true
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "REMOVE_CRL",
  "result": {}
}
```

Метод RANDOM_BYTES

Метод призначено для генерації криптографічно стійкої псевдовипадкової послідовності.

Структура поля parameters у запиті

Назва поля	Тип	Опис
length	Integer	Довжина даних у байтах, значення повинно бути більше 0

Структура поля result у відповіді

Назва поля	Тип	Опис
bytes	Base64	Згенеровані випадкові дані

Приклад запиту:

```
{
  "method": "RANDOM_BYTES",
  "parameters": {
    "length": 32
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "RANDOM_BYTES",
  "result": {
    "bytes": "exW+I6VE8nAnKS7U+xtLxePF1V5dJXIIV2RevNW5LmM="
  }
}
```

Метод DIGEST

Метод призначено для гешування даних. Для вказання алгоритму гешування використовується параметри hashAlgo або signAlgo (одночасно тільки один із них). Дані для гешування можуть бути задані одним з трьох способів:

- безпосередньо у вигляді base64-кодованої строки;
- повним шляхом до файлу, що містить дані для гешування (файл повинен бути доступним для читання);
- вказівником та розміром на області пам'яті.

Структура поля parameters у запиті

Назва поля	Тип	Опис
hashAlgo	String	Алгоритм гешування. Взаємовиключний до поля signAlgo
signAlgo	String	Алгоритм підпису. Взаємовиключний до поля hashAlgo
bytes	Base64	Вхідні дані, якщо дані задані безпосередньо
file	String	Вхідні дані, що зберігаються у файлі
ptr	Hex	Покажчик на дані в оперативній пам'яті
size	Integer	Довжина вхідних даних в оперативній пам'яті

Структура поля result у відповіді

Назва поля	Тип	Опис
hashAlgo	String	Алгоритм гешування, який був використаний
bytes	Base64	Значення геш-функції від даних

Приклад запиту, дані задані безпосередньо:

```
{
  "method": "DIGEST",
  "parameters": {
    "hashAlgo": "2.16.840.1.101.3.4.2.1",
    "bytes": "VGhlIHF1aWNrIGJyb3duIGZve...IGRvZw=="
  }
}
```

Приклад запиту, дані задані що зберігаються у файлі:

```
{
  "method": "DIGEST",
  "parameters": {
    "hashAlgo": "2.16.840.1.101.3.4.2.1",
    "file": "~/docs/filename.doc"
  }
}
```

Приклад запиту, дані знаходяться в пам'яті:

```
{
  "method": "DIGEST",
  "parameters": {
    "hashAlgo": "2.16.840.1.101.3.4.2.1",
    "ptr": "000001940D2F8400",
    "size": 10000000
  }
}
```

Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "DIGEST",
  "result": {
    "hashAlgo": "2.16.840.1.101.3.4.2.1",
    "bytes": "16j7swfXgJRpypq8sAguT41...vzfJ5ZI="
  }
}
```

Метод ASN1_DECODE

Метод призначено для декодування DER-кодованих ASN.1 даних. Перелік ASN.1-типів, що можуть бути декодовані наведено в Додатку Г.

Структура поля parameters у запиті

Назва поля	Тип	Опис
items	Object[], DECODE_ITEM[]	Масив структур, що містять дані для декодування

Структура DECODE_ITEM

Назва поля	Тип	Опис
bytes	Base64	Дані для декодування
id	String	Ідентифікатор даних. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
decoded	Object[], DECODED_ITEM[]	Масив структур, що містить декодовані дані

Структура DECODED_ITEM

Назва поля	Тип	Опис
tag	String	Ідентифікатор ASN.1-типу (тег)
value	Base64 Boolean String	Декодоване значення відповідно ASN.1-типу
integer	Integer	Ціле число. Опціональний
bytes	Base64	Значення без декодування. Опціональний

Приклад запиту:

```
{
  "method": "ASN1_DECODE",
  "parameters": {
    "items": [
      {
        "id": "boolean-true",
        "bytes": "AQH/"
      }, {
        "id": "integer-1",
        "bytes": "AgEB"
      }, {
        "id": "integer-big",
        "bytes": "AhQ9tz578NV1sgEAAAABAAAAugAAAA=="
      }, {
        "id": "octet-string",
        "bytes": "BAowMTIzNDU2Nzg5"
      }, {
        "id": "null",
        "bytes": "BQA="
      }, {
        "id": "oid-12345",
        "bytes": "BgQqAwQF"
```

```
      }, {
        "id": "printable-string",
        "bytes": "EytUaGUgcXVpY2sgYnJvd24gZm94IGp1bXBzIG92
                  ZXIgdGhlIGxhenkgZG9n"
      }, {
        "id": "utf8-string",
        "bytes": "DFzQodC/0YDQuNGC0L3QsCDQsdGD0YDQsCDQu9C4
                  0YHQuNGG0Y8g0YHRgtGA0LjQsdCw0ZQg0YfQtdGA
                  0LXQtyDQu9C10LTQsNGH0L7Qs9C+INGB0L7QsdCw0LrRgw=="
      }, {
        "id": "ia5-string",
        "bytes": "FitUaGUgcXVpY2sgYnJvd24gZm94IGp1bXBzIG92
                  ZXIgdGhlIGxhenkgZG9n"
      },
      {
        "id": "utc-time",
        "bytes": "Fw0yMTA3MDgxMjM0NTZa"
      }, {
        "id": "generalized-time",
        "bytes": "GA8yMDIxMDcwODEyMzQ1Nlo="
      }
    ]
  }
}
```

Приклад відповіді:

```
{
  "errorCode": 0,
  "method": "ASN1_DECODE",
  "result": {
    "decoded": [
      {
        "id": "boolean-true",
        "tag": "BOOLEAN",
        "value": true
      }, {
        "id": "integer-1",
        "tag": "INTEGER",
        "value": "AQ==",
        "integer": 1
      }, {
        "id": "integer-big",
        "tag": "INTEGER",
        "value": "Pbc+e/DVdbIBAAAAAQAAALoAAAA="
      }, {
        "id": "octet-string",
        "tag": "OCTET_STRING",
        "value": "MDEyMzQ1Njc4OQ=="
      }, {
        "id": "null",
        "tag": "NULL"
      },
      {
        "id": "oid-12345",
        "tag": "OID",
        "value": "1.2.3.4.5"
      }, {
        "id": "printable-string",
        "tag": "PRINTABLE_STRING",
```

```
        "value": "The quick brown fox jumps over the lazy dog",
        "bytes": "VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVy
                  IHRoZSBsYXp5IGRvZw=="
      }, {
        "id": "utf8-string",
        "tag": "UTF8_STRING",
        "value": "Спритна бура лисиця стрибає через ледачого собаку",
        "bytes": "0KHQv9GA0LjRgtC90LAg0LHRg9GA0LAg0LvQuNGB
                  0LjRhtGPINGB0YLRgNC40LHQsNGUINGH0LXRgNC1
                  0Lcg0LvQtdC00LDRh9C+0LPQviDRgdC+0LHQsNC60YM="
      }, {
        "id": "ia5-string",
        "tag": "IA5_STRING",
        "value": "The quick brown fox jumps over the lazy dog",
        "bytes": "VGhlIHF1aWNrIGJyb3duIGZveCBqdW1wcyBvdmVy
                  IHRoZSBsYXp5IGRvZw=="
      }, {
        "id": "utc-time",
        "tag": "UTC_TIME",
        "value": "2021-07-08 12:34:56",
        "integer": 1625747696000
      }, {
        "id": "generalized-time",
        "tag": "GENERALIZED_TIME",
        "value": "2021-07-08 12:34:56",
        "integer": 1625747696000
      }
    ]
  }
}
```

Метод ASN1_ENCODE

Метод призначено для кодування даних згідно DER-кодування ASN.1. Перелік ASN.1-типів, що можуть бути кодовані наведено в Додатку Г.

Структура поля parameters у запиті

Назва поля	Тип	Опис
items	Object[], ENCODE_ITEM[]	Масив структур, що містить дані для кодування

Структура ENCODE_ITEM

Назва поля	Тип	Опис
tag	String	Ідентифікатор ASN.1-типу (тег)
value	Base64 Boolean String	Дані для кодування. Опціональний, залежить від типу
integer	Integer	Ціле число. Опціональний, залежить від типу
id	String	Ідентифікатор даних. Опціональний

Структура поля result у відповіді

Назва поля	Тип	Опис
encoded	Object[], ENCODED_ITEM[]	Масив структур, що містить кодовані дані

Структура ENCODED_ITEM

Назва поля	Тип	Опис
bytes	Base64	Кодовані дані
id	String	Ідентифікатор даних. Опціональний

Приклад запиту:

```
{
  "method": "ASN1_ENCODE",
  "parameters": {
    "items": [
      {
        "id": "boolean-FALSE",
        "tag": "BOOLEAN",
        "value": false
      }, {
        "id": "boolean-TRUE",
        "tag": "BOOLEAN",
        "value": true
      }, {
        "id": "integer-1-as-integer",
        "tag": "INTEGER",
        "integer": 1
      }, {
        "id": "integer-2-as-value",
        "tag": "INTEGER",
        "value": "Ag=="
      }, {
        "id": "integer-big",
```

```
        "tag": "INTEGER",
        "value": "Pbc+e/DVdbIBAAAAAQAAALoAAAA="
      }, {
        "id": "octet-string",
        "tag": "OCTET_STRING",
        "value": "MDEyMzQ1Njc4OQ=="
      }, {
        "id": "null",
        "tag": "NULL"
      },
      {
        "id": "oid-12345",
        "tag": "OID",
        "value": "1.2.3.4.5"
      }, {
        "id": "printable-string",
        "tag": "PRINTABLE_STRING",
        "value": "The quick brown fox jumps over the lazy dog"
      }, {
        "id": "utf8-string",
        "tag": "UTF8_STRING",
        "value": "Спритна бура лисиця стрибає через ледачого собаку"
      }, {
        "id": "ia5-string",
        "tag": "IA5_STRING",
        "value": "The quick brown fox jumps over the lazy dog"
      }, {
        "id": "utc-time-as-unixtime",
        "tag": "UTC_TIME",
        "integer": 1625747696000
      }, {
        "id": "utc-time-as-text",
        "tag": "UTC_TIME",
        "value": "2021-07-08 12:34:56"
      }, {
        "id": "generalized-time-as-unixtime",
        "tag": "GENERALIZED_TIME",
        "integer": 1625747696000
      }, {
        "id": "generalized-time-as-text",
        "tag": "GENERALIZED_TIME",
        "value": "2021-07-08 12:34:56"
      }
    ]
  }
}
```


Приклад відповіді:


```
{
  "errorCode": 0,
  "method": "ASN1_ENCODE",
  "result": {
    "encoded": [ {
        "id": "boolean-FALSE",
        "bytes": "AQEA"
      }, {
        "id": "boolean-TRUE",
        "bytes": "AQH/"
      }, {
```

```
      "id": "integer-1-as-integer",
      "bytes": "AgEB"
    }, {
      "id": "integer-2-as-value",
      "bytes": "AgEC"
    }, {
      "id": "integer-big",
      "bytes": "AhQ9tz578NV1sgEAAAABAAAAugAAAA=="
    }, {
      "id": "octet-string",
      "bytes": "BAowMTIzNDU2Nzg5"
    }, {
      "id": "null",
      "bytes": "BQA="
    }, {
      "id": "oid-12345",
      "bytes": "BgQqAwQF"
    }, {
      "id": "printable-string",
      "bytes": "EytUaGUgcXVpY2sgYnJvd24gZm94IGp1bXBzIG92
               ZXIgdGhlIGxhenkgZG9n"
    }, {
      "id": "utf8-string",
      "bytes": "DFzQodC/0YDQuNGC0L3QsCDQsdGD0YDQsCDQu9C4
               0YHQuNGG0Y8g0YHRgtGA0LjQsdCw0ZQg0YfQtdGA
               0LXQtyDQu9C10LTQsNGH0L7Qs9C+INGB0L7QsdCw0LrRgw=="
    }, {
      "id": "ia5-string",
      "bytes": "FitUaGUgcXVpY2sgYnJvd24gZm94IGp1bXBzIG92
               ZXIgdGhlIGxhenkgZG9n"
    }, {
      "id": "utc-time-as-unixtime",
      "bytes": "Fw0yMTA3MDgxMjM0NTZa"
    }, {
      "id": "utc-time-as-text",
      "bytes": "Fw0yMTA3MDgxMjM0NTZa"
    }, {
      "id": "generalized-time-as-unixtime",
      "bytes": "GA8yMDIxMDcwODEyMzQ1Nlo="
    }, {
      "id": "generalized-time-as-text",
      "bytes": "GA8yMDIxMDcwODEyMzQ1Nlo="
    } ]
  }
}
```

Додаток А. Коди помилок

Таблиця А.1. Коди помилок

Код	Значення	Опис
RET_OK	0	Операція виконана успішно
RET_UAPKI_GENERAL_ERROR	0x1001	Невизначена помилка
RET_UAPKI_CONNECTION_ERROR	0x1002	Помилка з'єднання з сервером
RET_UAPKI_INVALID_JSON_FORMAT	0x1003	Неправильний формат JSON запиту
RET_UAPKI_INVALID_METHOD	0x1004	Метод не існує
RET_UAPKI_INVALID_PARAMETERS	0x1005	Неправильний параметр
RET_UAPKI_UNKNOWN_PROVIDER	0x1006	Невідомий провайдер
RET_UAPKI_FILENAME_REQUIRED	0x1007	Потребує ім'я файлу, як ідентифікатор НКІ
RET_UAPKI_LOGIN_REQUIRED	0x1008	Потребує ім'я користувача
RET_UAPKI_NOT_INITIALIZED	0x1009	Бібліотеку не ініціалізовано
RET_UAPKI_ALREADY_INITIALIZED	0x100A	Бібліотеку вже ініціалізовано
RET_UAPKI_NO_STORAGE	0x100B	НКІ не відкрито
RET_UAPKI_KEY_NOT_SELECTED	0x100C	Ключ не вибрано
RET_UAPKI_INVALID_KEY_USAGE	0x100D	Ключ не може бути використаний для операції за призначенням
RET_UAPKI_UNSUPPORTED_ALG	0x100E	Криптографічний примітив не підтримується
RET_UAPKI_INVALID_HASH_SIZE	0x100F	Неправильний розмір геш-значення
RET_UAPKI_INVALID_KEY_ID	0x1010	Неправильний ідентифікатор ключа
RET_UAPKI_JSON_FAILURE	0x1011	Помилка підсистеми роботи з JSON
RET_UAPKI_INVALID_BIT_STRING	0x1012	Помилка ASN.1
RET_UAPKI_UNEXPECTED_BIT_STRING	0x1013	Помилка ASN.1
RET_UAPKI_TOO_LONG_BIT_STRING	0x1014	Помилка ASN.1
RET_UAPKI_TIME_ERROR	0x1015	Неправильний час
RET_UAPKI_NOT_SUPPORTED	0x1016	Не підтримується
RET_UAPKI_NOT_ALLOWED	0x1017	Заборонено
RET_UAPKI_OFFLINE_MODE	0x1018	Спроба виконати операцію, для виконання якої не вистачає інформації (потрібен доступ до онлайн-ресурсів НЕДП), при увімкненому режимі offline. Наприклад, відсутні актуальні СВС
RET_UAPKI_STORAGE_NOT_OPEN	0x1019	НКІ не відкрито
RET_UAPKI_PROVIDER_NOT_LOADED	0x101A	Бібліотеку роботи з НКІ не завантажено
RET_UAPKI_UNSUPPORTED_CMAPI	0x101B	Бібліотека роботи з НКІ не підтримує дану операцію
RET_UAPKI_FILE_OPEN_ERROR	0x1020	Помилка відкриття файлу
RET_UAPKI_FILE_READ_ERROR	0x1021	Помилка читання файлу
RET_UAPKI_FILE_WRITE_ERROR	0x1022	Помилка запису файлу
RET_UAPKI_FILE_GET_SIZE_ERROR	0x1023	Помилка визначення розміру файлу
RET_UAPKI_FILE_DELETE_ERROR	0x1024	Помилка видалення файлу
RET_UAPKI_HTTP_STATUS_NOT_OK	0x1025	Відповідь сервера не успішна
RET_UAPKI_INVALID_CONTENT_INFO	0x1030	Неправильна структура підпису або зашифрованих даних
RET_UAPKI_INVALID_STRUCT	0x1031	Неправильна (або не підтримується) структура ASN.1
RET_UAPKI_INVALID_STRUCT_VERSION	0x1032	Неправильний (або не підтримується) номер версії структури ASN.1

RET_UAPKI_CONTENT_NOT_PRESENT	0x1033	Відсутні дані для перевірки підпису
RET_UAPKI_INVALID_ATTRIBUTE	0x1034	Неправильні атрибути підпису
RET_UAPKI_ATTRIBUTE_NOT_PRESENT	0x1035	Відсутні атрибути підпису
RET_UAPKI_EXTENSION_NOT_PRESENT	0x1036	Відсутні необхідні розширення
RET_UAPKI_EXTENSION_NOT_SET_CRITICAL	0x1037	Відсутні розширення не визначені як критичні
RET_UAPKI_INVALID_COUNT_ITEMS	0x1038	Неправильна кількість елементів
RET_UAPKI_INVALID_DIGEST	0x1039	Неправильний геш
RET_UAPKI_OTHER_RECIPIENT	0x103A	Файл зашифровано для іншого отримувача
RET_UAPKI_CERT_STORE_LOAD_ERROR	0x1040	Помилка завантаження кешу сертифікатів
RET_UAPKI_CERT_NOT_FOUND	0x1041	Сертифікат не знайдено
RET_UAPKI_CERT_VALIDITY_NOT_BEFORE_ERROR	0x1042	Термін дії сертифікату не настав
RET_UAPKI_CERT_VALIDITY_NOT_AFTER_ERROR	0x1043	Термін дії сертифікату закінчився
RET_UAPKI_CERT_ISSUER_NOT_FOUND	0x1044	Сертифікат видавця не знайдено
RET_UAPKI_CERT_STATUS_REVOKED	0x1045	Сертифікат відкликано
RET_UAPKI_CERT_STATUS_UNKNOWN	0x1046	Статус сертифікату не визначено
RET_UAPKI_CRL_STORE_LOAD_ERROR	0x1050	Помилка завантаження кешу СВС
RET_UAPKI_CRL_URL_NOT_PRESENT	0x1051	Відсутня точка розповсюдження СВС
RET_UAPKI_CRL_NOT_DOWNLOADED	0x1052	Помилка завантаження СВС
RET_UAPKI_CRL_NOT_FOUND	0x1053	СВС не знайдено
RET_UAPKI_CRL_EXPIRED	0x1054	Термін чинності СВС закінчився
RET_UAPKI_OCSP_URL_NOT_PRESENT	0x1060	Відсутня точка доступу до OCSP
RET_UAPKI_OCSP_NOT_RESPONDING	0x1061	Сервер OCSP не відповідає
RET_UAPKI_OCSP_RESPONSE_NOT_SUCCESSFUL	0x1062	Відповідь OCSP не успішна
RET_UAPKI_OCSP_RESPONSE_VERIFY_FAILED	0x1063	Недійсний підпис відповіді OCSP
RET_UAPKI_OCSP_RESPONSE_VERIFY_ERROR	0x1064	Помилка при валідації підпису відповіді OCSP
RET_UAPKI_OCSP_RESPONSE_INVALID_NONCE	0x1065	Поле nonce відповіді та запиту OCSP не співпадають
RET_UAPKI_OCSP_RESPONSE_INVALID	0x1066	Структура відповіді OCSP не правильна
RET_UAPKI_TSP_URL_NOT_PRESENT	0x1070	Не вказана точка доступу до серверу TSP
RET_UAPKI_TSP_NOT_RESPONDING	0x1071	Сервер TSP не відповідає
RET_UAPKI_TSP_RESPONSE_NOT_GRANTED	0x1072	Позначку часу не надано
RET_UAPKI_TSP_RESPONSE_NOT_EQUAL_REQUEST	0x1073	Відповідь TSP не співпадає з запитом
RET_UAPKI_TSP_RESPONSE_INVALID	0x1074	Структура відповіді TSP не правильна

Таблиця А.2. Коди помилок роботи з НКІ

Код	Значення	Опис
RET_OK	0	Операція виконана успішно
RET_CM_GENERAL_ERROR	0x0401	Невизначена помилка провайдера НКІ
RET_CM_INVALID_PARAMETER	0x0402	Неправильний параметр
RET_CM_LIBRARY_NOT_LOADED	0x0403	Помилка завантаження провайдера НКІ
RET_CM_ALREADY_INITIALIZED	0x0404	Провайдер НКІ вже ініціалізовано

RET_CM_NOT_INITIALIZED	0x0405	Провайдер НКІ не ініціалізовано
RET_CM_UNSUPPORTED_API	0x0406	Функція не підтримується провайдером НКІ
RET_CM_UNSUPPORTED_PARAMETER	0x0407	Параметр не підтримується провайдером НКІ
RET_CM_NO_SESSION	0x0408	Сесію роботи з НКІ не відкрито
RET_CM_INVALID_MECHANISM	0x0409	Неправильний ідентифікатор криптографічного примітиву
RET_CM_UNSUPPORTED_MAC	0x040A	Непідтримуваний алгоритм розрахунку імітовставки
RET_CM_INVALID_MAC	0x040B	Неправильна імітовставка
RET_CM_WITHOUT_MAC	0x040C	Відсутня імітовставка
RET_CM_INVALID_CONTENT_INFO	0x040D	Неправильна структура підпису або зашифрованих даних
RET_CM_UNSUPPORTED_CONTENT_INFO	0x040E	Неправильна (або не підтримується) структура ASN.1
RET_CM_INVALID_SAFE_BAG	0x040F	Неправильна структура зашифрованих даних або ключів
RET_CM_NOT_AUTHORIZED	0x0410	Користувача не авторизовано
RET_CM_INVALID_PASSWORD	0x0411	Неправильний пароль
RET_CM_READONLY_SESSION	0x0412	НКІ відкрито тільки для читання
RET_CM_BAG_NOT_FOUND	0x0413	Неправильна структура ASN.1
RET_CM_KEY_NOT_FOUND	0x0414	Ключ не знайдено на НКІ
RET_CM_CERTIFICATE_NOT_FOUND	0x0415	Сертифікат не знайдено на НКІ
RET_CM_KEY_NOT_SELECTED	0x0416	Поточний ключ не вибрано
RET_CM_UNSUPPORTED_ALG	0x0417	Криптографічний примітив не підтримується
RET_CM_UNSUPPORTED_CIPHER_ALG	0x0418	Алгоритм шифрування не підтримується
RET_CM_UNSUPPORTED_ELLIPTIC_CURVE	0x0419	Еліптична крива не підтримується
RET_CM_UNSUPPORTED_RSA_LEN	0x041A	Довжина ключа RSA не підтримується
RET_CM_UNSUPPORTED_KEY_DERIVATION_FUNC_ALG	0x041B	Функція вироблення ключа не підтримується
RET_CM_INVALID_HASH	0x041C	Неправильне геш-значення
RET_CM_INVALID_KEY	0x041D	Неправильний ключ
RET_CM_INVALID_ELLIPTIC_CURVE	0x041E	Неправильна еліптична крива
RET_CM_INVALID_UTF8_STR	0x041F	Неправильна строка UTF8
RET_CM_INVALID_JSON	0x0420	Неправильний формат JSON
RET_CM_INVALID_PARAM_DH	0x0421	Неправильні параметри протоколу узгодження ключа
RET_CM_UNSUPPORTED_KEY_CONTAINER	0x0422	Непідтримуваний формат ключа
RET_CM_UNSUPPORTED_FORMAT	0x0423	Непідтримуваний формат
RET_CM_CONNECTION_ERROR	0x0424	Помилка з’єднання
RET_CM_INVALID_RESPONSE	0x0425	Неправильна відповідь
RET_CM_RESPONSE_ERROR	0x0426	Неправильна відповідь
RET_CM_ACCESS_DENIED	0x0427	Доступ заборонено
RET_CM_JSON_FAILURE	0x0428	Помилка підсистеми роботи з JSON
RET_CM_STORAGE_NOT_OPEN	0x0429	Немає відкритого НКІ
RET_CM_TOKEN_ERROR	0x042A	Помилка апаратного НКІ
RET_CM_TOKEN_NO_FREE_SESSIONS	0x042B	Апаратний НКІ зайнятий
RET_CM_TOKEN_NO_FREE_SPACE	0x042C	На апаратному НКІ немає вільного місця
RET_CM_TOKEN_ALREADY_LOGGED	0x042D	Користувача вже автентифіковано
RET_CM_STORAGE_NOT_FOUND	0x042F	НКІ не знайено
RET_CM_FILE_OPEN_ERROR	0x0430	Помилка відкриття файлу

RET_CM_FILE_READ_ERROR	0x0431	Помилка зчитування файлу
RET_CM_FILE_WRITE_ERROR	0x0432	Помилка запису файлу
RET_CM_FILE_DELETE_ERROR	0x0433	Помилка видалення файлу
RET_CM_DECODE_ASN1_ERROR	0x0434	Помилка декодування ASN.1
RET_CM_ENCODE_ASN1_ERROR	0x0435	Помилка DER-кодування ASN.1
RET_CM_PASSWORD_NOT_SET	0x0436	Пароль не встановлено
RET_CM_INVALID_CERTIFICATE	0x0437	Неправильний сертифікат
RET_CM_INVALID_KEYID	0x0438	Неправильний ідентифікатор ключа
RET_CM_INVALID_WRAPPED_KEY	0x0439	Неправильний або пошкоджений захищений ключ

Додаток Б. Перелік форматів підпису

Таблиця Б.1. Перелік форматів підпису, що підтримуються бібліотекою

Назва формату	Короткий опис
RAW	Вихідна послідовність даних ЕП. Має специфічний бінарний формат для кожного алгоритму електронного підпису.
CMS	Базовий формат підпису з ідентифікацією підписувача за ідентифікатором відкритого ключа. Має два обов'язкових підписаних атрибути: 1) contentType; 2) messageDigest.
CAdES-BES	Базовий формат підпису з ідентифікацією підписувача за ідентифікатором сертифікату підписувача. Має три обов'язкових підписаних атрибутів: 1) contentType; 2) messageDigest; 3) signingCertificateV2.
CAdES-T	Формат підпису, який є розширеним варіантом формату CAdES-BES із двома додатковими атрибутами: позначкою часу від даних (contentTimestamp) і позначкою часу від підпису (timeStampToken). Має 4 обов'язкових підписаних атрибутів: 1) contentType; 2) messageDigest; 3) signingCertificateV2; 4) contentTimestamp. Має один обов'язковий непідписаний атрибут: 1) timeStampToken.
CAdES-C	Формат підпису, який є розширеним варіантом формату CAdES-T із двома додатковими непідписаними атрибутами: certificateRefs і revocationRefs. Перелік 4 обов'язкових підписаних атрибутів збігається з CAdES-T. Має три обов'язкових непідписаних атрибути: 1) timeStampToken; 2) certificateRefs; 3) revocationRefs. Для перевірки статусу сертифікатів використовується тільки СВС, відповідно в атрибуті revocationRefs знаходяться посилання на використані СВС.
CAdES-XL CAdES-LT	Формат підпису, який є розширеним варіантом формату CAdES-C із двома додатковими непідписаними атрибутами: certValues і revocationValues. Перелік 4 обов'язкових підписаних атрибутів збігається з CAdES-T. Має 5 обов'язкових непідписаних атрибутів: 1) timeStampToken; 2) certificateRefs; 3) revocationRefs; 4) certValues; 5) revocationValues.
CAdES-A	Формат підпису, який є розширеним варіантом формату CAdES-XL з одним

CAdES-LTA	додатковим непідписаним атрибутом — архівною позначкою часу archiveTimestampV3. Перелік 4 обов'язкових підписаних атрибутів збігається з CAdES-T. Має 6 обов'язкових непідписаних атрибутів: 1) timeStampToken; 2) certificateRefs; 3) revocationRefs; 4) certValues; 5) revocationValues; 6) archiveTimestampV3.

Формати сімейства CMS/CAdES дозволяють використання необов'язкових атрибутів. Наприклад, у підписаних атрибутах використовують атрибут signingTime. В залежності від завдання можна використовувати нестандартні атрибути (в рамках стандарту CMS/CAdES).

Назви формату підпису "CAdES-LT" та "CAdES-LTA" є синонімами "CAdES-XL" та "CAdES-A" відповідно. Вони можуть використовуватися тільки в методі SIGN.

Додаток В. Перелік розширень сертифікату

Таблиця В.1. Перелік розширень сертифікату, які можуть бути декодованими в CERT_INFO

Назва розширення	OID	Короткий опис
authorityInfoAccess	1.3.6.1.5.5.7.1.1	Інформація про ресурси видавника
authorityKeyIdentifier	2.5.29.35	Ідентифікатор ключа видавця сертифікату
basicConstraints	2.5.29.19	Основні обмеження
cRLDistributionPoints	2.5.29.31	Посилання на адреси зберігання повних СВС-файлів
certificatePolicies	2.5.29.32	Політики сертифікату
freshestCRL	2.5.29.46	Посилання на адреси зберігання часткових СВС-файлів
keyUsage	2.5.29.15	Призначення ключа
qcStatements	1.3.6.1.5.5.7.1.3	Профілі кваліфікованого сертифікату
subjectDirectoryAttributes	2.5.29.9	Додаткові атрибути підписувача
subjectInfoAccess	1.3.6.1.5.5.7.1.11	Інформація про ресурси доступні власнику сертифікату
subjectKeyIdentifier	2.5.29.14	Ідентифікатор ключа власника сертифікату

Таблиця В.2. Перелік значень розширення KEY_USAGE

Назва поля	Тип	Опис
digitalSignature	Boolean	Опціональний, за замовчанням false
contentCommitment	Boolean	Синонім nonRepudiation. Опціональний, за замовчанням false
keyEncipherment	Boolean	Опціональний, за замовчанням false
dataEncipherment	Boolean	Опціональний, за замовчанням false
keyAgreement	Boolean	Опціональний, за замовчанням false
keyCertSign	Boolean	Опціональний, за замовчанням false
crlSign	Boolean	Опціональний, за замовчанням false
encipherOnly	Boolean	Опціональний, за замовчанням false
decipherOnly	Boolean	Опціональний, за замовчанням false

Додаток Г. Перелік ASN1-типів

Таблиця Г.1. Перелік ASN1-типів, що підтримуються в методах ASN1_DECODE та ASN1_ENCODE

Назва ASN1-типу	Короткий опис
BOOLEAN	Булеве значення
INTEGER	Ціле число
OCTET_STRING	Довільна послідовність байт (октетів)
NULL	Відсутнє значення
OID	Ідентифікатор об'єкта
PRINTABLE_STRING	Текст у кодуванні ASCII підмножини призначеного для друку
UTF8_STRING	Текст у кодуванні UTF-8
IA5_STRING	Текст у кодуванні IA-5 (перша частина ASCII таблиці)
UTC_TIME	Всесвітній координований час без століття (рік задається 2 цифрами)
GENERALIZED_TIME	Всесвітній координований час із століттям (рік задається 4 цифрами)

Додаток Ґ. Перелік елементів опису сертифікату

Таблиця Ґ.1. Перелік елементів опису сертифікату в структурі RDNAME_INFO *

Назва	OID	Distinguished name
C	2.5.4.6	country
CN	2.5.4.3	commonName
G	2.5.4.42	givenName
L	2.5.4.7	locality
O	2.5.4.10	organization
OI	2.5.4.97	organizationIdentifier
OU	2.5.4.11	organizationalUnit
S	2.5.4.8	state
SERIALNUMBER	2.5.4.5	serialNumber
SN	2.5.4.4	surname
STREET	2.5.4.9	streetAddress
TITLE	2.5.4.12	title

* якщо елемент опису сертифікату невідомий (відсутній в таблиці), то замість назви використовується значення OID.