ТОВ «С.І.Т»

UAPKI

бібліотека

Настанова програміста

Версія 2.0.7

Київ, 2022

Історія версій

Дата	Версія	Опис
02.08.2021	2.0.1	Початковий реліз
16.08.2021	2.0.2	У відповіді методу додана назва методу
01.09.2021	2.0.3	Зміни у параметрах методів INIT та VERIFY_CERT
16.09.2021	2.0.4	Зміни у способах конфігурації бібліотеки. Для сертифікатів і CRL створені окремі ідентифікатори (оновлен інтерфейс методів, які використовують їх). Додана підтримка пакету сертифікатів в методі ADD_CERT
11.11.2021	2.0.5	Зміни в методах OPEN, DEGIST, SIGN, VERIFY, VERIFY_CERT
17.11.2021	2.0.6	Зміни в методу CERT_INFO
07.11.2022	2.0.7	Додані методи DECRYPT та ENCRYPT

Загальні відомості

Цей документ призначено для використання розробниками додатків, які мають потреби у інтеграції механізмів електронного цифрового підпису.

До складу бібліотеки входять бінарні файли (бібліотеки), які представлені в таблиці 1 (префікси і розширення залежать від платформи/ОС) і файл конфігурації.
Таблиця 1. Перелік бінарних файлів бібліотеки

№	Базова назва бінарного файлу	Короткий опис
1	uapki	Головна бібліотека, що реалізує основну логіку роботи
2	uapkic	Бібліотека криптографії, обов'язкова
3	uapkif	Бібліотека форматів, обов'язкова
4	cm-<storage-name> cm-pkcs12	Бібліотека провайдера сховищ ключів. Наприклад для файлового сховища це буде “cm-pkcs12”. Необхідна, якщо використовуються функції, які залежать від приватних ключів (наприклад, підпис даних)

Бібліотека реалізовує свій функціонал за допомогою 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 використання текстової строки, що складається за правилами JSON (далі JSON-строка).

Таблиця 2. Перелік методів бібліотеки

Назва методу	Короткий опис
VERSION	Версія бібліотеки
INIT	Ініціалізація бібліотеки
DEINIT	Завершення роботи бібліотеки (де-ініціалізація)
PROVIDERS	Перелік провайдерів сховищ ключів
STORAGES	Перелік сховищ ключів доступних через обраний провайдер
STORAGE_INFO	Інформація про сховище ключів
OPEN	Відкрити сховище ключів
CLOSE	Закрити сховище ключів
KEYS	Перелік ключів у відкритому сховищі
SELECT_KEY	Вибрати ключ
CREATE_KEY	Створити ключ
DELETE_KEY	Видалити ключ
GET_CSR	Отримати запит на сертифікат
CHANGE_PASSWORD	Зміна паролю (PIN коду) до сховища ключів
INIT_KEY_USAGE	Ініціалізація використання ключа
SIGN	Підпис даних
VERIFY	Перевірка підписаних даних

ENCRYPT	Шифрування даних
DECRYPT	Розшифрування даних
ADD_CERT	Додати сертифікат до кешу сертифікатів
CERT_INFO	Інформація з сертифіката
GET_CERT	Отримати сертифікат із кешу сертифікатів
LIST_CERTS	Перелік сертифікатів у кешу сертифікатів
REMOVE_CERT	Видалити сертифікат із кешу сертифікатів
VERIFY_CERT	Перевірка сертифікату
ADD_CRL	Додати CRL до кешу CRL
CRL_INFO	Інформація з CRL
DIGEST	Гешування даних
ASN1_DECODE	Декодування DER-кодованих ASN1 даних
ASN1_ENCODE	Кодування даних згідно DER-кодування ASN1

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);

Опис функції **process**:
параметр request – це вказівник на JSON-строку, яка описує параметри методу (назва методу, параметри методу);
повертає вказівник на JSON-строку, що зберігає результат виконання методу. Цей вказівник після обробки повинен завжди видаляться функцією json_free.

Опис функції **json_free**:
параметр result – це вказівник на текстовий рядок у форматі JSON, який був отриманий виконанням функції process;
нічого не повертає.

Опис методів бібліотеки

Всі запити і відповіді методів мають єдиний формат, вони розрізнюються в полях parameters й result.

Формат запиту

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

Формат відповіді

Назва поля	**Тип**	**Опис**
errorCode	Integer	Код помилки
method	String	Назва методу
result	Object	Структура, що містить результат виконання методу
error	String	Короткий текстовий опис помилки. Опціональний

Метод VERSION

Метод призначений для визначення версії бібліотеки. Параметри в запиті відсутні.

Приклад запиту:

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

Структура поля result у відповіді

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

Приклад результату:

```
{
  "errorCode": 0,
  "method": "VERSION",
  "result": {
    "name": "UAPKI",
    "version": "2.0.5"
  }
}
```

Метод INIT

Метод призначений 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 використанні параметри за замовченням. Для роботи із сховищем ключів необхідно задати параметри CM-провайдерів.

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 CM-провайдерів. Опціональний
certCache	Object CERT_CACHE_PARAMS	Параметри кешу сертифікатів. Опціональний
crlCache	Object CRL_CACHE_PARAMS	Параметри кешу CRL. Опціональний
offline	Boolean	Режим роботи “офлайн”. Опціональний
tsp	Object TSP_PARAMS	Параметри TSP-сервісу. Опціональний

Структура CERT_CACHE_PARAMS

Назва поля	**Тип**	**Опис**
path	String	Повний шлях до каталогу. Опціональний
trustedCerts	Base64[]	Масив довірених сертифікатів. Опціональний

Структура CRL_CACHE_PARAMS

Назва поля	**Тип**	**Опис**
path	String	Повний шлях до каталогу. Опціональний

Структура TSP_PARAMS

Назва поля	Тип	Опис
url	String	URL-адреса TSP-сервісу. Опціональний
policyId	String	OID-ідентифікатор політики. Опціональний

Структура CMPROVIDERS_PARAMS

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

Структура CMPROVIDER_PARAMS

Назва поля	Тип	Опис
lib	String	Назва файлу бібліотеки СМ-провайдеру
config	Object	Параметри спецефічні для кожного СМ-провайдеру. Опціональний

Приклад файлу конфігурації:

```
{
  "cmProviders": {
    "dir": "C:/uapki/cm-libs/",
    "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,
  "tsp": {
    "url": "http://url_ca/services/tsp/",
    "policyId": "1.2.3.4"
  }
}
```

Приклад запиту з параметрами:

```
{
  "method": "INIT",
  "parameters": {
    "cmProviders": {
      "dir": "C:/uapki/cm-libs/",
      "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,
    "tsp": {
      "url": "http://url_ca/services/tsp/",
      "policyId": "1.2.3.4"
    }
  }
}
```

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
certCache	Object CERT_CACHE_INFO	Інформація о стані кешу сертифікатів (структура CERT_CACHE_INFO)
crlCache	Object CRL_CACHE_INFO	Інформація о стані кешу CRL (структура CRL_CACHE_INFO)
countCmProviders	Integer	Кількість підключених CM-провайдерів
offline	Boolean	Режим роботи “офлайн”
tsp	Object TSP_INFO	Інформація о параметрах TSP-сервісу (структура TSP_INFO)

Структура CERT_CACHE_INFO

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

Структура CRL_CACHE_INFO

Назва поля	Тип	Опис
countCrls	Integer	Кількість CRL у кешу CRL

Структура TSP_INFO

Назва поля	Тип	Опис
url	String	URL-адреса TSP-сервісу. Опціональний
policyId	String	OID-ідентифікатор політики. Опціональний

Приклад результату:

```
{
  "errorCode": 0,
  "method": "INIT",
  "result": {
    "certCache": {
      "countCerts": 29,
      "countTrustedCerts": 5
    },
    "crlCache": {
      "countCrls": 4
    },
    "countCmProviders": 2,
    "offline": false,
    "tsp": {
      "url": "http://url_ca/services/tsp/",
      "policyId": "1.2.3.4"
    }
  }
}
```

Метод DEINIT

Метод призначений для звільнення ресурсів бібліотеки які були ініціалізовані в методі INIT. Параметри в запиті відсутні. Результат — пуста структура.

Приклад запиту:

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

Приклад результату:

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

Метод PROVIDERS

Метод призначений для отримання переліку провайдерів. Параметри в запиті відсутні.

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
providers	Array of Object	Масив структур PROVIDER_INFO

Структура PROVIDER_INFO

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

Приклад результату:

```
{
  "errorCode": 0,
  "method": "PROVIDERS",
  "result": {
    "providers": [{
      "id": "DIAMOND",
      "apiVersion": "1.0.0",
      "libVersion": "1.0.0",
      "description": "DIAMOND-provider",
      "manufacturer": "2021 SPECINFOSYSTEMS LLC",
      "supportListStorages": true
    }, {
      "id": "PKCS12",
      "apiVersion": "1.0.0",
      "libVersion": "1.0.0",
      "description": "PKCS12-provider",
      "manufacturer": "2021 SPECINFOSYSTEMS LLC"
      "supportListStorages": false
    }]
  }
}
```

Метод STORAGES

Метод призначений для отримання переліку сховищ провайдера.
Провайдери можуть не підтримувати даний метод: PKCS12- та CLOUD-провайдер не надають перелік доступних сховищ.

Структура поля parameters у запиті

Назва поля	Тип	Опис
provider	String	Ідентифікатор провайдера

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
storages	Array of Object	Масив структур з описом сховища

Приклад результату, якщо метод підтримується:

```
{
  "errorCode": 0,
  "method": "STORAGES",
  "storages": [{
    "id": "000001",
    "manufacturer": "SPECINFOSYSTEMS LLC",
    "description": "DIAMOND 1000 token",
    "serial": "000001",
    "label": "User 1"
    },
    { ... }
  ]
}
```

Приклад результату, якщо метод не підтримується:

```
{
  "errorCode": 258,
  "method": "STORAGES",
  "result": {
    "message": "PKCS12-provider not supported API
      'provider_storage_info'."
  }
}
```

Метод STORAGE_INFO

Метод призначений для отримання інформації про сховище провайдеру.
Провайдери можуть не підтримувати даний метод: PKCS12- та CLOUD-провайдер не надають перелік доступних сховищ.

Структура поля parameters у запиті

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

Приклад запиту:

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

Приклад результату, якщо метод не підтримується:

```
{
  "errorCode": 258,
  "method": "STORAGE_INFO",
  "result": {
    "message": "PKCS12-provider not supported API
               'provider_storage_info'."
  }
}
```

Метод OPEN

Відкриття сховища здійснюється за допомогою методу OPEN. Має три обов'язкових параметра ("provider", "storage" й "mode") та специфічні параметри, які залежать від типу провайдера сховища.

Режими роботи із сховищем ("mode")

Значення	Тип
"RW"	Доступні всі методи для роботи з ключами
"RO"	Доступні методи для роботи з ключами, які не змінюють сховище (аналогія режиму "тільки на читання" для звичайних файлів)
"CREATE"	Створення нового сховища, доступні всі методи для роботи з ключами

Структура поля parameters у запиті до CLOUD-провайдера

Назва поля	Тип	Опис
provider	String	Ідентифікатор CLOUD-провайдера
storage	String	URL-адреса сховища
mode	String	Режим роботи із сховищем. Опціональний (значення за замовчанням: "RW")
username	String	Ім'я облікового запису користувача
password	String	Пароль до облікового запису користувача

Приклад запиту до CLOUD-провайдера (відкриття сховища в режимі тільки на читання):

```
{
  "method": "OPEN",
  "parameters": {
    "provider": "CLOUD",
    "storage": "http://url-storage",
    "mode": "RO",
    "username": "username",
    "password": "password"
  }
}
```

Структура поля parameters у запиті до PKCS12-провайдера

Назва поля	Тип	Опис
provider	String	Ідентифікатор PKCS12-провайдера
storage	String	Ім'я файлу сховища
mode	String	Режим роботи із сховищем. Опціональний (значення за замовчанням: "RW")
password	String	Пароль до файлового сховища
openParams	Object	Параметри для створення нового файлового сховище

Приклад запиту до PKCS12-провайдера (створення нового файлового сховища):

```
{
  "method": "OPEN",
  "parameters": {
    "provider": "PKCS12",
```

```
    "storage": "file.p12",
    "mode": "CREATE",
    "password": "password",
    "openParams": {
      "bagCipher": "2.16.840.1.101.3.4.1.42",
      "bagKdf": "1.2.840.113549.2.9",
      "iterations": 10000,
      "macAlgo": "2.16.840.1.101.3.4.2.1"
    }
  }
}
```

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
mechanisms	Array of Object	Масив структур з описом механізмів, які доступні для використання
userPresense	Boolean	Опціональний

Приклад результату:

```
{
  "errorCode": 0,
  "method": "OPEN",
  "result": {
    "description": "PKCS12",
    "manufacturer": "2021 SPECINFOSYSTEMS LLC",
    "label": "file.p12",
    "model": "PKCS12",
    "serial": "file.p12",
    "mechanisms": [{
      "id": "1.2.804.2.1.1.1.1.3.6",
      "name": "DSTU-4145",
      "keyParam": ["1.2.804.2.1.1.1.1.3.1.1.2.6", ... ],
      "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

Закриття сховища здійснюється за допомогою методу CLOSE. Параметри в запиті відсутні. Результат — пуста структура.

Приклад запиту:

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

Приклад результату:

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

Метод KEYS

Метод призначений для отримання переліку ключів на відкритому сховищі. Параметри в запиті відсутні.

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
keys	Array of Object	Масив структур KEY_INFO з описом ключів

Структура KEY_INFO

Назва поля	Тип	Опис
id	String	Ідентифікатор ключа. Унікальне значення
mechanismId	String	Ідентифікатор алгоритму ключа
parameterId	String	Ідентифікатор параметру ключа
signAlgo	Array of String	Ідентифікатори алгоритмів підпису, що підтримуються ключем
label	String	Текстове позначення ключа. Опціональний

Приклад результату:

```
{
  "errorCode": 0,
  "method": "KEYS",
  "result": {
    "keys": [{
      "id": "112233445566...DDEEFF00",
      "mechanismId": "1.2.804.2.1.1.1.1.3.1",
      "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"
    },
    {
      "id": "112233445566...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"
    }
    ]
  }
}
```

Метод SELECT_KEY

Метод призначений для вибору поточного ключа у відкритому сховищі.

Структура поля parameters у запиті

Назва поля	Тип	Опис
id	String	Ідентифікатор ключа. Унікальний

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
signAlgo	Array of String	Ідентифікатори алгоритмів підпису, що підтримуються ключем
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 із сховища. Опціональний
extAuth	String	Параметри додаткової автентифікації. Опціональний

Приклад результату:

```
{
  "errorCode": 0,
  "method": "SELECT_KEY",
  "result": {
    "signAlgo": ["1.2.804.2.1.1.1.1.3.1.1", ... ],
    "certId": "MIH+MIHlMQsw ... NQAAAFwAAAA=",
    "cert": "MIIErjCCBFagAwIBAgIUFXeRu ... NcYCFp23iPeya2s="
  }
}
```

Метод CREATE_KEY

Метод призначений для створення нового ключа у відкритому сховищі. Параметри за якими може бути створений ключ визначаються при відкритті сховища. Якщо параметри не вказані, то буде створений ключ з параметрами за замовчанням: алгоритм ECDSA, параметр P256. Коли метод виконаний успішно, то новий ключ стає поточним ключем у сховищі.

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
mechanismId	String	Ідентифікатор алгоритму ключа. Опціональний
parameterId	String	Ідентифікатор параметру ключа. Опціональний
label	String	Текстове позначення ключа. Опціональний

Приклад запиту:

```
{
  "method": "CREATE_KEY",
  "parameters": {
    "mechanism": "1.2.804.2.1.1.1.1.3.1",
    "parameter": "1.2.804.2.1.1.1.1.3.1.1.2.6",
    "label": "new DSTU4145-key, M257_PB"
  }
}
```

Структура поля result у відповіді

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

Приклад результату:

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

Метод DELETE_KEY

Метод призначений для видалення ключа у відкритому сховищі. Результат — пуста структура.

Структура поля parameters у запиті

Назва поля	Тип	Опис
id	Hex	Ідентифікатор ключа. Унікальний

Приклад запиту:

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

Приклад результату:

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

Метод GET_CSR

Метод призначений для отримання запиту на формування сертифікату для поточного 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

Приклад запиту:

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

Структура поля result у відповіді

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

Приклад результату:

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

Метод CHANGE_PASSWORD

Метод призначений для зміни пароля у відкритому сховищі. Результат — пуста структура.

Структура поля parameters у запиті

Назва поля	Тип	Опис
password	String	Старий пароль
newPassword	String	Новий пароль

Приклад запиту:

```
{
  "method": "CHANGE_PASSWORD",
  "parameters": {
    "password": "123pass",
    "newPassword": "newpass"
  }
}
```

Приклад результату:

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

Метод INIT_KEY_USAGE

Метод призначений для ініціалізації використання ключа. Параметри запиту і результату залежать від сховища.

Приклад запиту:

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

Приклад результату:

```
{
    "result": {
        ...
    },
    "errorCode": 0
}
```

Метод SIGN

Метод призначений для підпису даних.

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
keyParams	Object, KEY_PARAMS	Набір параметрів сховища ключа. Опціональний

Структура SIGN_PARAMS

Назва поля	Тип	Опис
signatureFormat	String	Формат підпису: - “RAW” – вихідна послідовність даних ЕЦП - “CMS” – базовий з ідентифікацією підписувача за ідентифікатором відкритого ключа - “CAdES-BES” – базовий ЕЦП
signAlgo	OID	Ідентифікатор алгоритму підпису. Опціональний
digestAlgo	OID	Ідентифікатор алгоритму гешування. Опціональний
detachedData	Boolean	Зовнішній підпис (дані не інкапсулюються). Опціональний, по замовчанню true
includeCert	Boolean	Додати до підпису сертифікат власника ключа. Опціональний, по замовчанню false
includeTime	Boolean	Додати до підпису час хосту (недовірений). Опціональний, по замовчанню false
includeContentTS	Boolean	Додати до підпису позначку часу від даних. Опціональний, по замовчанню false
certs	[]	Масив сертифікатів. Опціональний. (зарезервовано для майбутнього використання)

Структура DATA_TBS_PARAMS

Назва поля	Тип	Опис
id	String	Ідентифікатор даних
bytes	Base64	Дані для підпису
isDigest	Boolean	Тип даних для підпису. Якщо true, то поле bytes містить геш, інакше оригінальні дані. 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 KEY_PARAMS

Назва поля	**Тип**	**Опис**
permission	Base64	* Дозвіл на використання. Опціональний
provider	String	Ідентифікатор провайдера. Опціональний
storage	String	Ідентифікатор сховища. Опціональний
keyId	Hex	Ідентифікатор ключа. Опціональний
username	String	Ім'я облікового запису користувача хмарного сховища. Опціональний
password	String	Пароль до сховища ключа. Опціональний
PIN	String	PIN-код доступу до ключа. Опціональний
OTP	String	Одноразовий код для доступу до віддаленого ключа. Опціональний
serial	String	Серійний номер пристрою. Опціональний (зарезервовано для майбутнього використання)
tokenLabel	String	зарезервовано для майбутнього використання
keyLabel	String	зарезервовано для майбутнього використання

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": "0J3QsNCx0L7RgCDQt ... B1YXBraS4="
    }, {
      "id": "doc-1",
      "bytes": "VHJpcGxlIENyb3duIG9mIE1vdG9yc3BvcnQ="
    }]
  }
}
```

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 результату:

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

Метод VERIFY

Метод призначений для верифікації підписаних даних у форматі “CMS/CAdES” або в криптографічному виді (“RAW”). Поле “signature” є обов'язковим — в ньому підписані дані зберігаються в полі “signature.bytes”.
Для верифікації 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.
Для верифікації RAW-формату підпису необхідно більше параметрів:
1) поле “signParams” містить параметри підпису (“signAlgo” - обов'язково);
2) поле “signer” 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 містять підписані дані
signParams	Object SIGN_PARAMS	Набір параметрів підпису. Умовно-опціональний
signer	Object SIGNER_PUBKEY	Набір параметрів підписувача. Умовно-опціональний
options	Object OPTIONS	Додаткові параметри. Опціональний

Структура SIGNATURE_DATA

Назва поля	**Тип**	**Опис**
bytes	Base64	Підписані дані
content	Base64	Оригінальні дані. Умовно-опціональний
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 OPTIONS

Назва поля	Тип	Опис
encodeCert	Boolean	Кодувати значення сертифікат у Base64
validateCertByOCSP	Boolean	Перевірити статус сертифікату за OCSP. (зарезервовано для майбутнього використання)
validateCertByCRL	Boolean	Перевірити статус сертифікату за CRL. (зарезервовано для майбутнього використання)

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

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

Приклад запиту для верифікації 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"
    },
    "signer": {
      "certId": "MIH+MIHlMQsw ... NQAAAFwAAAA="
    }
  }
}
```

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

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

Назва поля	Тип	Опис
statusSignature	String	Статус цифрового 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	Ідентифіктор сертифікату надписувача
status	String	Статус підписаних даних: "INDETERMINATE", "TOTAL-FAILED", "TOTAL-VALID"
statusSignature	String	Статус цифрового підпису: "FAILED", "INVALID", "VALID"
statusMessageDigest	String	Статус цифрового дайджесту даних: "FAILED", "INVALID", "VALID"
statusEssCert	String	Статус ідентифікації сертифікату підписника (опціональний): "UNDEFINED", "NOT PRESENT", "FAILED", "INVALID", "VALID"
signingTime	Time	Час підпису. Опціональний
signedAttributes	Object[]	Масив атрибутів у структурі ATTRIBUTE_PARAMS, що зберігаються у полі signedAttributes
unsignedAttributes	Object[]	Масив атрибутів у структурі ATTRIBUTE_PARAMS, що зберігаються у полі unsignedAttributes. Опціональний
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

Структура TIMESTAMP_INFO

Назва поля	Тип	Опис
genTime	Time	Позначка часу
policyId	OID	Ідентифікатор політики TSP
statusDigest	String	Статус значення позначки часу: "FAILED", "INVALID", "VALID"
statusSign	String	Статус перевірки підпису в позначці часу: "FAILED", "INVALID", "VALID"
hashAlgo	OID	Ідентифікатор алгоритму гешування. Опціональний — присутній, якщо поле “statusDigest” не дорівнює "VALID"
hashedMessage	Base64	Значення гешу. Опціональний — присутній, якщо поле “statusDigest” не дорівнює "VALID"

Приклад результату верифікації підписаних даних у “RAW”-форматі:


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


Приклад результату верифікації підписаних даних у “CMS/CAdES”-форматі:


```
{
  "errorCode": 0,
  "method": "SIGN",
  "result": {
    "content": {
      "type": "1.2.840.113549.1.7.1",
      "bytes": "QWxpY2UgYW5k...ZV9hbmRfQm9i"
    },
    "certIds": [ "MGwwVDELMAkG ... CwAAAD0AAAA=", ... ],
    "signatureInfos": [{
      "signerCertId": "MGwwVDELMAkG ... CwAAAD0AAAA=",
      "status": "TOTAL-VALID",
      "statusSignature": "VALID",
      "statusMessageDigest": "VALID",
      "statusEssCert": "VALID",
      "signingTime": "2021-07-08 12:32:39",
      "contentTS": {
        "genTime": "2021-07-08 12:32:39",
        "policyId": "1.2.804.2.1.1.1.2.3.1",
        "statusDigest": "VALID"
      },
      "signedAttributes": [{
        "type": "1.2.840.113549.1.9.3",
        "bytes": "BgkqhkiG9w0BBwE="
      }, {
        "type": "1.2.840.113549.1.9.5",
        "bytes": "Fw0yMTA3MDgwOTMyMzla"
      }, ...]
    }]
  }
}
```

Метод ENCRYPT

Метод призначений для шифрування даних.

Структура поля parameters у запиті

Назва поля	Тип	Опис
content	Object CONTENT_PARAMS	Структура з даними і параметрами шифрування
originatorCertIds	Base64[]	Масив ідентифікаторів сертифікатів відправника. Опціональний.
recipientInfos	Object[], RECIPINFO_PARAMS	Масив структур, що містять параметри отримувачів
unprotectedAttrs	Object[], ATTRIBUTE_PARAMS[]	Масив структур, що містять 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 сертифіката отримувача
kdfAlgo	OID	Ідентифікатор алгоритму функції формування ключа
keyWrapAlgo	OID	Ідентифікатор алгоритму обгортання 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	"1.2.804.2.1.1.1.1.1.3.3.2"	"1.2.804.2.1.1.1.1.3.7"	"1.2.804.2.1.1.1.1.1.3.11"
2	"1.2.804.2.1.1.1.1.1.3.3.2"	"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"

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"
    }]
  }
}
```

Структура поля result у відповіді

Назва поля	**Тип**	**Опис**
bytes	Base64	Зашифровані дані

Приклад результату:

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

Метод DECRYPT

Метод призначений для розшифрування даних.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	Зашифровані дані

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
content	Object CONTENT_INFO	Структура CONTENT_INFO, що містить розшифровані дані
originatorCertId	Base64	Ідентифікатор сертифіката 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 результату:

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

Метод ADD_CERT

Метод призначений для додавання сертифікатів до кешу сертифікатів. Сертифікати додаються двома способами: массивом сертифікатів або (одним) пакетом сертифікатів (p7b-файл). 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).

Структура поля parameters у запиті

Назва поля	**Тип**	**Опис**
certificates	Base64[]	Масив сертифікатів. Есклюзивний до поля bundle
bundle	Base64	Пакет сертифікатів. Есклюзивний до поля certificates
permanent	Boolean	Ознака зберегти сертифікат у кешу сертифікатів

Приклад запиту:

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

Структура поля result у відповіді

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

Структура CERT_ADDED

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

Приклад результату:

```
{
  "errorCode": 0,
  "method": "ADD_CERT",
  "result": {
    "added": [{
      "certId": "MIH+MIHlMQsw ... NQAAAFwAAAA=",
      "isUnique": true
    }, {
      "certId": "MIGFMG0xCzAJ ... AAAAhAAAAA==",
      "isUnique": false
    },
    ... ]
  }
}
```

Метод CERT_INFO

Метод призначений для отримання інформації, яка зберігається в сертифікаті. Сертифікат повинен бути відповідним стандарту x.509 та мінімальну версію 3.
Метод повертає масив розширень в тому порядку в якому вони зберігаються в сертифікаті.

Структура поля parameters у запиті

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

Приклад запиту:

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

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 розширеній які має сертифікат
signatureInfo	Object SIGNATURE_INFO	Цифровий підпис сертифікату
selfSigned	Boolean	Ознака, що сертифікат самопідписаний

Структура SUBJECT_PUBLICKEY_INFO

Назва поля	Тип	Опис
bytes	Base64	DER-кодоване поле subjectPublicKeyInfo
algorithm	String	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	Недекодоване 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	String	Ідентифікатор алгоритму підпису
parameters	Base64	Параметри алгоритму підпису. Опціональний
signature	Base64	Значення підпису

Приклад результату:


```
{
  "errorCode": 0,
  "method": "CERT_INFO",
  "result": {
    "bytes": "MIIErjCCBFagAwIBAg...cYCFp23iPeya2s=",
    "version": 3,
    "serialNumber": "157791B9508857ED04000000...0000",
    "issuer": {
      "C": "UA",
      "SERIALNUMBER": "UA-12345678-0001",
      "CN": "Центр сертифікації ключів",
      "O": "ТОВ Магія",
      "OU": "ЦСК",
      "L": "Київ"
    },
    "validity": {
      "notBefore": "2020-08-26 12:34:56",
      "notAfter": "2022-08-26 12:34:56"
    },
    "subject": {
      "C": "UA",
      "CN": "Серпень Аугусто",
      "L": "Марсополіс",
      "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": "D0069AA0A8DF7D707A...28CC7"
          }
        }
      },
      {
        "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 призначений 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

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
bytes	Base64	Сертифікат

Приклад результату:

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

Метод LIST_CERTS

Метод призначений для отримання переліку ідентифікаторів сертифікатів із кешу сертифікатів. Підтримує пагінацію.

Структура поля parameters у запиті

Назва поля	Тип	Опис
offset	Integer	Індекс першого сертифікату. Опціональний (значення за замовчанням: 0)
pageSize	Integer	Максимальна кількість сертифікатів. Опціональний

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис
certIds	Base64[]	Масив ідентифікаторів сертифікатів
count	Integer	Кількість сертифікатів
offset	Integer	Індекс першого сертифікату
pageSize	Integer	Максимальна кількість сертифікатів

Приклад результату:

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

Метод REMOVE_CERT

Метод видаляє сертифікат якщо він є в кеші сертифікатів, інакше повертається помилка - сертифікат не знайден ("CERT_NOT_FOUND"). Результат — пуста структура.

Структура поля parameters у запиті

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

Приклад запиту:

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

Структура поля result у відповіді

Назва поля	Тип	Опис

Приклад результату:

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

Метод VERIFY_CERT

Метод призначений для верифікації сертифікату. Якщо сертифікат самопідписаний, то поле issuerCertId у відповіді відсутнє.
Поле validateTime встановлює значення часу за який треба визначити валідність сертифікату за CRL. Якщо це поле відсутнє, то використовується поточний час хосту.

Структура поля parameters у запиті

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

Приклад запиту:

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

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 цифрового підпису: "VALID", "INVALID", "FAILED"
issuerCertId	Base64	Ідентифікатор сертифікату видавця. Опціональний
validateByCRL	Object VALIDATE_BY_CRL	Результат перевірки сертифікату користувача з використанням CRL. Опціональний
validateByOCSP	Object VALIDATE_BY_OCSP	Результат перевірки сертифікату користувача з використанням OCSP. Опціональний

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 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 CRL
delta	Object CRL_INFO	Інформація про частковий CRL. Опціональний

Структура поля CRL_INFO

Назва поля	Тип	Опис
url	String	URL зберігання CRL. Опціональний
crlId	Base64	Ідентифікатор CRL в CRL-кеші
statusSignature	String	Статус цифрового підпису: "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	Hex	Ідентифікатор ключа CA

producedAt	Time	Час створення OCSP-відповіді
thisUpdate	Time	Час створення поточного запису OCSP
nextUpdate	Time	Час створення наступного запису OCSP

Приклад результату:


```
{
  "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="
  }
}
```

Метод ADD_CRL

Метод призначений для додавання CRL до кешу CRL. Повертає ідентифікатор CRL у CRL-кеші.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	CRL
permanent	Boolean	Ознака зберегти CRL у кешу CRL

Приклад запиту:

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

Структура поля result у відповіді

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

Приклад результату:

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

Метод CRL_INFO

Метод призначений для отримання інформації, яка зберігається в CRL (списку відкликаних сертифікатів). CRL повинен бути відповідним стандарту x.509.

Структура поля parameters у запиті

Назва поля	Тип	Опис
bytes	Base64	CRL

Приклад запиту:

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

Структура поля result у відповіді

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

Структура поля REVOKED_CERT_INFO

Назва поля	Тип	Опис
userCertificate	Hex	Серійний номер викликанного 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 результату:


```
{
  "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 DIGEST

Метод призначений для гешування даних. Для вказання алгоритму гешування використовується параметри hashAlgo або signAlgo (одночасно тільки один із двох). Дані для гешування могут бути задані або безпосередньо у вигляді base64-кодованної строки, або вказанням імені файлу, або вказання на область пам’яті.

Структура поля parameters у запиті для верифікації цифрового підпису

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

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’яти:

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

Структура поля result у відповіді

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

Приклад результату:


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

Метод ASN1_DECODE

Метод призначений для декодування DER-кодованих ASN1 даних. Перелік ASN1-типів, що можуть бути декодовані дано в додатку 3.

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

Приклад запиту:

```
{
  "method": "ASN1_DECODE",
  "parameters": {
    "items": [{
      "bytes": "BAowMTIzNDU2Nzg5",
      "id": "octet-10bytes"
    }, {
      "bytes": "AgEB",
      "id": "integer-1"
    }, {
      "bytes": "AgMBAAE=",
      "id": "integer-65537"
    }, {
      "bytes": "AhQ9tz578NV1sgEAAAABAAAAugAAAA==",
      "id": "integer-big"
    }, {
      "bytes": "AQH/"
    }, {
      "bytes": "BQA="
    }, {
      "bytes": "BgQqAwQF"
    }, {
      "bytes": "EwJVQQ=="
    }, {
      "bytes": "ExdFeGFtcGxlOiBBbGljZSBhbmQgQm9iLg=="
    }, {
      "bytes": "DCXQn9GA0LjQutC70LDQtDog0JDQu9GW0YHQsCDRliDQkdC+0LEu"
    }, {
      "bytes": "Fw0yMTA3MDgxMjM0NTZa"
    }, {
      "bytes": "GA8yMDIxMDcwODEyMzQ1Nlo="
    }, {
      "bytes": "AwIGwA=="
    }]
  }
}
```

Структура поля result у відповіді

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

Структура DECODED_ITEM

Назва поля	Тип	Опис
tag	String	Ідентифікатор ASN1-типу (тег)
value	Base64 Boolean String	Декодоване значення відповідно ASN1-типу
integer	Integer	Ціле число. Опціональний
bytes	Base64	Значення без декодування. Опціональний

Приклад результату:


```
{
  "errorCode": 0,
  "method": "ASN1_DECODE",
  "result": {
    "decoded": [{
      "id": "octet",
      "tag": "OCTET_STRING",
      "value": "MDEyMzQ1Njc4OQ=="
    }, {
      "id": "integer-1",
      "tag": "INTEGER",
      "value": "AQ==",
      "integer": 1
    }, {
      "id": "integer-65537",
      "tag": "INTEGER",
      "value": "AQAB",
      "integer": 65537
    }, {
      "id": "integer-big",
      "tag": "INTEGER",
      "value": "Pbc+e/DvdbIBAAAAAQAAALoAAAA="
    }, {
      "tag": "BOOLEAN",
      "value": true
    }, {
      "tag": "NULL"
    }, {
      "tag": "OID",
      "value": "1.2.3.4.5"
    }, {
      "tag": "PRINTABLE_STRING",
      "value": "UA",
      "bytes": "VUE="
    }, {
      "tag": "PRINTABLE_STRING",
      "value": "Example: Alice and Bob.",
      "bytes": "RXhhbXBsZTogQWxpY2UgYW5kIEJvYi4="
    }, {
```

```
      "tag": "UTF8_STRING",
      "value": "Приклад: Аліса і Боб.",
      "bytes": "0J/RgNC40LrQu9Cw0LQ6INCQ0LvRltGB0LAg0ZYg0JHQvtCxLg=="
    }, {
      "tag": "UTC_TIME",
      "value": "2021-07-08 12:34:56",
      "integer": 1625747696000
    }, {
      "tag": "GENERALIZED_TIME",
      "value": "2021-07-08 12:34:56",
      "integer": 1625747696000
    }, {
      "tag": "BIT_STRING",
      "value": "wA==",
      "integer": 3
    }]
  }
}
```

Метод ASN1_ENCODE

Метод призначений для кодування даних згідно DER-кодування ASN1. Перелік ASN1-типів, що можуть бути кодовані дано в додатку 3.

Структура поля parameters у запиті

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

Структура ENCODE_ITEM

Назва поля	Тип	Опис
tag	String	Ідентифікатор ASN1-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

Приклад запиту:

```
{
  "method": "ASN1_ENCODE",
  "parameters": {
    "items": [{
      "id": "octet",
      "tag": "OCTET_STRING",
      "value": "MDEyMzQ1Njc4OQ=="
    }, {
      "id": "integer-1-as-integer",
      "tag": "INTEGER",
      "integer": 1
    }, {
      "id": "integer-2-as-value",
      "tag": "INTEGER",
      "value": "Ag=="
    }, {
      "id": "integer-65537-as-integer",
      "tag": "INTEGER",
      "integer": 65537
    }, {
      "id": "integer-big",
      "tag": "INTEGER",
      "value": "Pbc+e/DVdbIBAAAAAQAAALoAAAA="
    }, {
      "id": "null",
      "tag": "NULL"
    }, {
      "id": "oid",
      "tag": "OID",
      "value": "1.2.3.4.5"
    }]
  }
}
```

Структура поля result у відповіді

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

Структура ENCODED_ITEM

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

Приклад результату:


```
{
  "errorCode": 0,
  "method": "ASN1_ENCODE",
  "result": {
    "encoded": [{
      "id": "octet",
      "bytes": "BAowMTIzNDU2Nzg5"
    }, {
      "id": "integer-1-as-integer",
      "bytes": "AgEB"
    }, {
      "id": "integer-2-as-value",
      "bytes": "AgEC"
    }, {
      "id": "integer-65537-as-integer",
      "bytes": "AgMBAAE="
    }, {
      "id": "integer-big",
      "bytes": "AhQ9tz578NV1sgEAAAABAAAAugAAAA=="
    }, {
      "id": "null",
      "bytes": "BQA="
    }, {
      "id": "oid",
      "bytes": "BgQqAwQF"
    }]
  }
}
```

Додаток 1. Коди помилок

Таблиця. Коди помилок

Код	**Опис**
RET_OK	Операція виконана успішно
RET_UAPKI_GENERAL_ERROR	Невизначена помилка
RET_UAPKI_CONNECTION_ERROR	Помилка з'єднання з сервером
RET_UAPKI_INVALID_JSON_FORMAT	Неправильний формат JSON запиту
RET_UAPKI_INVALID_METHOD	Невалідний метод
RET_UAPKI_INVALID_PARAMETERS	Невалідний параметр
RET_UAPKI_UNKNOWN_PROVIDER	Невідомий провайдер
RET_UAPKI_FILENAME_REQUIRED	Потребує ім'я файлу як ідентифікатор сховища
RET_UAPKI_LOGIN_REQUIRED	Потребує ім'я користувача як ідентифікатор сховища
RET_UAPKI_NOT_INITIALIZED	Бібліотеку не ініціалізовано
RET_UAPKI_ALREADY_INITIALIZED	Бібліотеку вже ініціалізовано
RET_UAPKI_NO_STORAGE	Сховище не відкрито
RET_UAPKI_NO_KEY	Ключ не вибрано
RET_UAPKI_INVALID_KEY_USAGE	Ключ не може бути використаний для операції не за призначенням
RET_UAPKI_NOT_ALLOWED	Операція не дозволена

Додаток 2. Перелік розширень сертифікату, які можуть бути декодованими в CERT_INFO

Таблиця. 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	
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 CRL-файлів
certificatePolicies	2.5.29.32	Політики сертифікату
freshestCRL	2.5.29.46	Посилання на адреси зберігання часткових CRL-файлів
keyUsage	2.5.29.15	Призначення ключа
qcStatements	1.3.6.1.5.5.7.1.3	
subjectDirectoryAttributes	2.5.29.9	Додаткові атрибути підписувача
subjectInfoAccess	1.3.6.1.5.5.7.1.11	
subjectKeyIdentifier	2.5.29.14	Ідентифікатор ключа власника сертифікату

Додаток 3. Перелік ASN1-типів, що підтримуються в методах ASN1_DECODE, ASN1_ENCODE

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

Назва ASN1-типу	**Короткий опис**
`INTEGER`	Ціле число
`OCTET_STRING`	Довільна послідовність байт (октетів)
`NULL`	Нулеве значення
`OID`	Ідентифікатор об'єкту