Міністерство освіти і науки України
Національний Технічний Університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Фізико-технічний інститут

Лабораторна работа №2
з дисципліни
«Методи реалізації криптографічних
механізмів»
Тема:
« Вибір рішень для реалізації Web-сервісу електронного цифрового підпису »

Виконав
Студент групи ФІ-22мн
Русєв Денис
Перевірив
Кудін А.М.

Київ-2023

Для створення Web-сервісу ЕЦП мною було обрано мову програмування GO.

Go — це статично типізована скомпільована мова програмування високого рівня, розроблена в Google Робертом Ґріземером, Робом Пайком і Кеном Томпсоном . Він синтаксично схожий на C, але з безпекою пам’яті, збиранням сміття, структурною типізацією та паралельністю у стилі CSP . Його часто називають Golang через його колишнє доменне ім’я golang.org, але його справжня назва Go.

Сервіс ЕЦП буде собою представляти Web API з двома контролерами POST Sign, POST Verify.

Використовується у веб-розробці — містить, як правило, певний набір HTTP-запитів, а також визначення структури HTTP-відповідей, для вираження яких найчастіше використовують XML або JSON формат, а також ProtoBuf, XDR і деякі інші. Web API є практично синонімом для веб-служби.

Контроллер Sign буде очікувати запит у форматі JSON, який повинен містити у собі данні про сховище для ключів, сертифікат та данні, які необхідно підписати :

{

Storage Storage

Certificate []byte

Data []byte

}

За допомогою геш-функції SHA-256(для геш-фунції завжди буде використовувати SHA-256.) буде отримано геш на данні, потім це геш-значення підписується за допомогою закритого ключа підписувача.

Після того як данні будуть успішно підписані буде сформований cms, який буде містити у собі файл, підпис та данні, які необхідні для перевірки підпису у майбутньому. У відповіді на запит буде повертатися підписаний файл у форматі JSON :

{

Cms []byte

}

Контролер Verify буде очікувати запит у форматі JSON, який повинен містити у собі підписаний файл:

{

Cms []byte

}

Отримавши необхідні данні з конверта підпису, знову обчислюється хеш та перевіряється на відповідність за допомогою відкритого ключа. Якщо хеш-значення та підпис збігаються, то перевірка пройшла успішно.

Після успішної перевірки підпису буде повернена відповідь у форматі JSON, що буде містити

данні, що були підписані

{

VerifiedData []byte

}

В інакшому випадку буду повернена відповідь у форматі JSON, що буде містити причину помилки.

{

ErrorReoson string

}