**Практическое задание № 9**

**Тема «Изучение стандартных средств для реализации симметричного и ассиметричного шифрование с использованием SubtleCrypto в js»**

Цель: Изучить интерфейс **SubtleCrypto**, Основные классы и структуры данных, разработать приложение для шифрования файлов использующих симметричные и ассиметричные алгоритмы шифрования

**Криптография в Js**

Web Crypto API - это интерфейс, позволяющий использовать криптографические примитивы для построения систем с использованием криптографии. Данный интерфейсвключают в себя возможность генерировать, использовать и применять пары криптографических ключей; шифровать и дешифровать сообщения; надежно генерировать случайные числа

Некоторые браузеры реализовали интерфейс под названием Crypto, без точной структуры. Чтобы избежать путаницы, методы и свойства этого интерфейса были удалены из браузеров, реализующих Web Crypto API, и все методы Web Crypto API доступны в новом интерфейсе: SubtleCrypto.

Интерфейс SubtleCrypto Web Crypto API предоставляет ряд низкоуровневых криптографических функций. Доступ к функциям SubtleCrypto осуществляется через объёкт Crypto.subtle .

Этот объект содержит набор методов для выполнения общих криптографических функций, таких как шифрование, хеширование, подписывание и генерация ключей. Поскольку все криптографические операции выполняются с необработанными двоичными данными, каждый метод SubtleCrypto имеет дело с типами ArrayBuffer и ArrayBufferView. Из-за того, что строки так часто становятся предметом криптографических операций, классы TextEncoder и TextDecoder будут часто использоваться вместе с SubtleCrypto для преобразования в строки и обратно.

**Хеширование**

Хеширование в SubtleCrypto представлено следующими алгоритмами.

* Sha1
* Sha256
* Sha385
* Sha512

Метод SubtleCrypto.digest() используется для создания хеша сообщения.

**Генерация ключей**

Генерация случайного CryptoKey выполняется с помощью метода SubtleCrypto.generateKey(algorithm, extractable, keyUsages). В этот метод передается объект params, указывающий целевой алгоритм, логическое значение,указывающее, должен ли ключ извлекаться из объекта CryptoKey, и массив строк — keyUsages, указывающий, с какими методами

Поскольку разные алгоритмы используют разный набор данных для ключей то в первый параметр содержит сооствутсвующее название алгоритма.

* RSA (RSASSA-PKCS1-v1\_5, RSA-PSS, or RSA-OAEP )использует объект RsaHashedKeyGenParams.
* ECDSA и ECDHи спользует объект EcKeyGenParams.
* HMAC использует объект HmacKeyGenParams.
* AES (AES-CTR, AES-CBC, AES-GCM, AES-KW) использует объект AesKeyGenParams

Значение extractable является логическим значением и указывает на возможность экспорта ключа.

Третий параметр keyUsages описывает, с какими алгоритмами можно использовать ключ.

* encrypt: Ключ используется для шифрования сообщений.
* decrypt: Ключ используется для расшифровки сообщений.
* sign: Ключ используется для подписи сообщений.
* verify: Ключ используется для проверки подписанного сообщения.
* deriveKey: Ключ используется для получения ключа.
* deriveBits: Ключ используется для получения битов.
* wrapKey: Ключ используется для упаковки ключа
* unwrapKey: Ключ используется для распоковки ключа.

**Создание цифровой подписи и проверка сообщений**

Объект SubtleCrypto позволяет использовать алгоритмы с открытым ключом для генерации подписей с использованием закрытого ключа или для проверки подписей с использованием открытого ключа. Они выполняются с использованием методов SubtleCrypto.sign() и SubtleCrypto.verify() соответственно. Для подписания сообщения требуется объект params, чтобы указать алгоритм и любые необходимые значения, частный CryptoKey и ArrayBuffer или ArrayBufferView для подписи.

**Упаковка и распаковка ключа**

Объект SubtleCrypto позволяет упаковывать и распаковывать ключи, чтобы обеспечить передачу по ненадежному каналу. Это выполняется с использованием методов SubtleCrypto.wrapKey() и SubtleCrypto.unwrapKey() соответственно. Для переноса ключа требуется строка форматирования, экземпляр CryptoKey для переноса, CryptoKey для выполнения переноса и объект params для указания алгоритма и любых необходимых значений.

**Задание к выполнению:**

1. Ознакомиться с созданием криптографического приложения
2. Выполнить генерацию и вывод в консоль случайный чисел.
3. Выполнить шифрование, дешифрование и хеширование своей фамилии по указанным алгоритмам.

|  |  |  |
| --- | --- | --- |
| № | Алгоритм шифрования(размер ключа) | Алгоритм хеширования |
| 6 | AES-CBC | SHA-256 |

Если количество человек в группе больше 15 то номер варианта определяется как x=n mod 15 где n номер в списке. В случае номер варианта в ходе решения уравнения ровен 0 то номер варианта равен 15.

1. Продемонстрировать упаковку и распаковку ключа, полученного в предыдущем задании используя алгоритм AES-KW.
2. Выполнить процедуру подписи сообщения и проверку подлинности с использованием RSA-PSS или ECDSA на выбор.

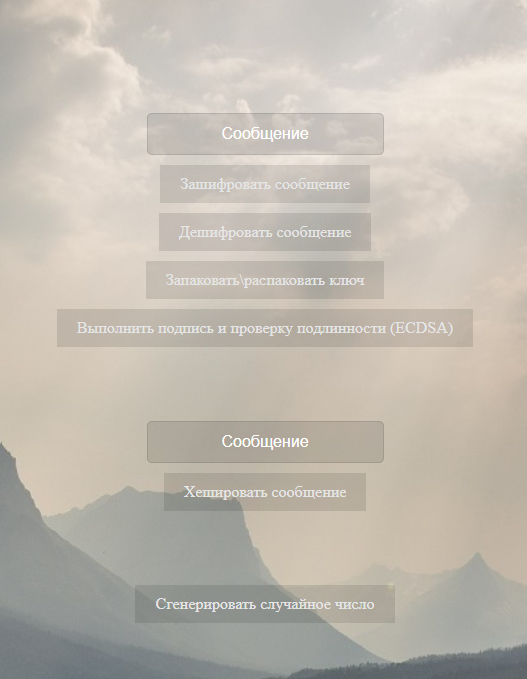


Рисунок – Вид страницы