МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образование «Белорусский государственный технологический университет»

**«**ИССЛЕДОВАНИЕ АЛГОРИТМОВ ГЕНЕРАЦИИ И ВЕРИФИКАЦИИ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ**»**

Студент: Бутурля Р.А.

ФИТ 3 курс 5 группа

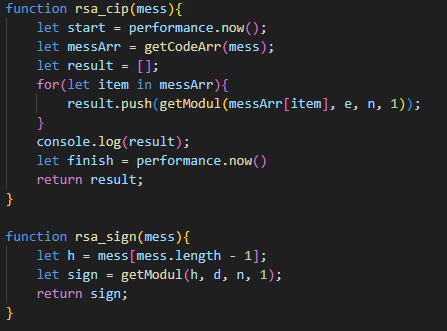
Вариант 3

Преподаватель: Савельева М. Г.

Минск 2023

1. Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться результатами выполнения предыдущих лабораторных работ, а также доступными библиотеками либо программными кодами.

Генерация RSA подписи происходит в методе rsa\_sign. Метод аналогичен методу из предыдущей лабораторной работы.



Листинг 1 – Метод rsa\_sign

После создания подписи она вместе с сообщением отсылается на сервер, который совершает проверку ЭЦП. Для этого был разработан метод verifyMess.

|  |
| --- |
| const verifyMess = (req, res) => {      let start = performance.now();      let {data} = req.body;      console.log(data);      let h = getModul(data.sign, data.e, 33, 1);      console.log(h);      let valid = false      if(h == data.messcip[data.messcip.length - 1])          valid = true;      let end = performance.now();      console.log('rsa - time : ' + (end-start));      res.end(JSON.stringify({sign:data.sign, valid: valid}));  } |

Листинг 2 – Метод verifyMess

Для генерации и верификации ЭЦП на основе алгоритмов Эль-Гамаля и Шнорра разработаны аналогичные RSA методы. Их листинг представлен ниже.

|  |
| --- |
| console.log('qwe');              let start = performance.now();              let m = document.getElementById('mess').value;              let k = document.getElementById('key').value;              let resArr = convertTo(m);              let mess = [];              let r = Math.pow(g, k)%p;              let e = (resArr[resArr.length - 1] \* Math.pow(db, k)) % p;              console.log(k, r, e, resArr[resArr.length - 1]);              mess.push({r, e}, resArr)              // }              let finish = performance.now(); |

Листинг 3 – Метод ag\_cip(Эль-Гамаль)

|  |
| --- |
| const verifyMessAG = (request, response) => {      // console.log('1');      // console.log(request.body.data);      let start = performance.now();      let isRes = false;      isRes = request.body.data[1][request.body.data[1].length - 1] == ag\_decip(request.body.data[0]) ? true : false;      let end = performance.now();      console.log('ag - time : ' + (end-start));      return response.end(JSON.stringify({mess:isRes}));      // return response.end(JSON.stringify({mess:false}));  } |

Листинг 4 – Метод verifyMessAG(Эль-Гамаль)

|  |
| --- |
| let start = performance.now();  let m = document.getElementById('mess').value;  let k = document.getElementById('key').value;  let resArr = convertTo(m);  let mess = [];  let r = Math.pow(g, k) % p;  let e = (k + x\*resArr[resArr.length - 1])%q;  mess.push({r, e}, resArr) |

Листинг 5 – Метод sh\_cip(Шнорр)

|  |
| --- |
| const verifyMessSh = (request, response) => {      let start = performance.now();      console.log('1');      console.log(request.body.data[1][request.body.data[1].length - 1], 'qqq');      let = isRes = request.body.data[0].e == ag\_decip(request.body.data[0])? true : false      let end = performance.now();      console.log('ag - time : ' + (end-start));      return response.end(JSON.stringify({mess:isRes}));      // return response.end(JSON.stringify({mess:false}));  } |

Листинг 4 – Метод verifyMessSH(Шнорр)

Для оценки времени построим таблицу зависимости времени от выбранного алгоритма.

Таблица 1 – Зависимость времени выполнения операции от длинны сообщения

|  |  |  |  |
| --- | --- | --- | --- |
| Длина сообщения \ алгоритм | rsa | Эль-Гамаля | Шнорра |
| 5 | 15.9 | 10.19 | 17.5 |

По результатам таблицы построим гистограмму – рисунок 1.

Рисунок 1 –Зависимость времени выбранного алгоритма подписи

**Вывод**

Изучил алгоритмы генерации и верификации электронной цифровой подписи и приобрел практические навыки их реализации.