Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №8»**

“Исследование ассиметричных шифров RSA и Эль-Гамаля”

**Выполнил:** студент 4 курса

1 группы специальности ИСИт

Палазник Арсений Викторович

**Проверил:** преподаватель

Сазонова Дарья Владимировна

Минск 2024

**Разработать авторское оконное приложение в соответствии с целью лабораторной работы. При этом можно воспользоваться доступными библиотеками либо программными кодами.**

**В основе вычислений – кодировочные таблицы Base64 и ASCII.**

**Приложение должно реализовывать следующие операции:**

* **зашифрование и расшифрование текстовых документов**

**на основе алгоритмов RSA и Эль-Гамаля;**

* **определение времени выполнения операций.**

Реализация приложения:

|  |
| --- |
| const crypto = require("crypto");  const nacl = require("tweetnacl");  nacl.util = require("tweetnacl-util");  // Генерация ключей RSA  function generateRSAKeys() {  const { publicKey, privateKey } = crypto.generateKeyPairSync("rsa", {  modulusLength: 2048,  publicKeyEncoding: { type: "pkcs1", format: "pem" },  privateKeyEncoding: { type: "pkcs1", format: "pem" },  });  return { publicKey, privateKey };  }  // Генерация ключей ElGamal  function generateElGamalKeys() {  const privateKey = nacl.box.keyPair();  return privateKey;  }  // RSA шифрование  function encryptRSA(plaintext, publicKey) {  return crypto.publicEncrypt(publicKey, Buffer.from(plaintext));  }  // RSA дешифрование  function decryptRSA(ciphertext, privateKey) {  return crypto.privateDecrypt(privateKey, ciphertext);  }  // ElGamal шифрование  function encryptElGamal(plaintext, publicKey, privateKey) {  const nonce = nacl.randomBytes(nacl.box.nonceLength);  const ciphertext = nacl.box(  nacl.util.decodeUTF8(plaintext),  nonce,  publicKey,  privateKey.secretKey  );  return { ciphertext, nonce };  }  // ElGamal дешифрование  function decryptElGamal(ciphertext, nonce, privateKey, publicKey) {  const decrypted = nacl.box.open(  ciphertext,  nonce,  publicKey,  privateKey.secretKey  );  if (!decrypted) {  throw new Error("Failed to decrypt ElGamal ciphertext");  }  return nacl.util.encodeUTF8(decrypted);  }  // Пример использования  (function main() {  console.log("Generating RSA keys...");  const rsaKeys = generateRSAKeys();  console.log("Generating ElGamal keys...");  const elgamalKeys = generateElGamalKeys();  const message = "Palanik Arseniy Viktorovich";  // RSA  console.log("\n--- RSA ---");  const rsaEncrypted = encryptRSA(message, rsaKeys.publicKey);  console.log("Encrypted (RSA):", rsaEncrypted.toString("base64"));  const rsaDecrypted = decryptRSA(rsaEncrypted, rsaKeys.privateKey);  console.log("Decrypted (RSA):", rsaDecrypted.toString("utf8"));  // ElGamal  console.log("\n--- ElGamal ---");  const { ciphertext, nonce } = encryptElGamal(  message,  elgamalKeys.publicKey,  elgamalKeys  );  console.log("Encrypted (ElGamal):", nacl.util.encodeBase64(ciphertext));  const elgamalDecrypted = decryptElGamal(  ciphertext,  nonce,  elgamalKeys,  elgamalKeys.publicKey  );  console.log("Decrypted (ElGamal):", elgamalDecrypted);  })(); |

Результат выполнения:

