Учреждение образования

«Белорусский государственный технологический университет»

**Кафедра информационных систем и технологий**

**«Отчёт по лабораторной работе №10»**

“Исследование алгоритмов генерации и верификации электронной цифровой подписи”

**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 результатами выполнения предыдущих лабораторных работ, а также доступными библиотеками либо программными кодами. Приложение должно реализовывать следующие операции:**

* **генерацию и верификацию ЭЦП на основе алгоритмов RSA, Эль-Гамаля и Шнорра;**
* **оценку времени выполнения указанных процедур при реальных (требуемых) ключевых параметрах**

Реализация приложения:

|  |
| --- |
| const crypto = require("crypto");  const { performance } = require("perf\_hooks");  // Utility to measure execution time  const measureTime = (label, fn) => {  const start = performance.now();  const result = fn();  const end = performance.now();  console.log(`${label} took ${(end - start).toFixed(3)}ms`);  return result;  };  // RSA Implementation  const rsa = {  generateKeys: () => {  const { publicKey, privateKey } = crypto.generateKeyPairSync("rsa", {  modulusLength: 2048,  });  return { publicKey, privateKey };  },  sign: (message, privateKey) => {  const hash = crypto.createHash("sha256").update(message).digest();  return crypto.privateEncrypt(privateKey, hash);  },  verify: (message, signature, publicKey) => {  const hash = crypto.createHash("sha256").update(message).digest();  const decryptedHash = crypto.publicDecrypt(publicKey, signature);  return hash.equals(decryptedHash);  },  };  // ElGamal Placeholder (Node.js doesn't have built-in support, requires library or custom implementation)  const elGamal = {  sign: () => {  console.log("ElGamal is not implemented in this sample.");  },  verify: () => {  console.log("ElGamal is not implemented in this sample.");  },  };  // Schnorr Placeholder (Requires custom implementation or library)  const schnorr = {  sign: () => {  console.log("Schnorr is not implemented in this sample.");  },  verify: () => {  console.log("Schnorr is not implemented in this sample.");  },  };  // Example Usage  const message = "Test message for digital signatures.";  // RSA Example  const rsaKeys = measureTime("RSA Key Generation", rsa.generateKeys);  const rsaSignature = measureTime("RSA Signing", () =>  rsa.sign(message, rsaKeys.privateKey)  );  const rsaVerification = measureTime("RSA Verification", () =>  rsa.verify(message, rsaSignature, rsaKeys.publicKey)  );  console.log("RSA Signature Verified:", rsaVerification);  // Note: Replace the ElGamal and Schnorr sections with appropriate libraries or implementations.  // Networking and key exchange can be added as per requirements using HTTP or WebSocket protocols. |

Результат выполнения:

