**1. Введение**

* 1. Назначение – 2
* 2. Определения, акронимы и сокращения – 2..4
* 3. Ссылки – 4
* 4. Краткий обзор – 4..

**2. Общее описание**

* 1. Взаимодействие продукта (с другими продуктами и компонентами)
* 2. Функции продукта (краткое описание)
* 3. Характеристики пользователя
* 4. Ограничения
* 5. Допущения и зависимости

**3. Детальные требования**

* 1. Требования к внешним интерфейсам  
  + 1. Интерфейсы пользователя
  + 2. Интерфейсы аппаратного обеспечения
  + 3. Интерфейсы программного обеспечения
  + 4. Интерфейсы взаимодействия
* 2. Функциональные требования
* 3. Требования к производительности
* 4. Проектные ограничения (и ссылки на стандарты)
* 5. Нефункциональные требования (надежность, доступность, безопасность и пр.)
* 6. Другие требования

**ВВЕДЕНИЕ**

* 1. **Назначение**

Была поставлена цель создать наиболее полную криптографическую библиотеку с реализацией на языке Java, улучшить существующие алгоритмы шифрования на Java, ускорить их производительность за счет внедрения новых средств последних редакций языка программирования.

Для достижения поставленной цели необходимо:

* Провести анализ существующих крипто-алгоритмов.
* Выбрать наиболее оптимальные из них с точки зрения криптостойкости и актуальности в настоящее время.
* По возможности усовершенствовать алгоритмы для увеличения производительности в режимах шифрования/дешифрования.
* Реализовать на языке программирования выбранные крипто-алгоритмы, собрав все в единую библиотеку.
* Создать подробную документацию.
* Сделать интерфейс взаимодействия написанной библиотеки с различными входными данными.
  1. **Определения, акронимы и сокращения**

**Защита информации** — комплекс технических, организационных и юридических мер, направленных на обеспечение целостности информации, на обеспечение возможности передать информацию, реализацию права на доступ к информации и множество других целей.

**Кодирование** — перевод из одного алфавита в другой с целью обеспечения удобной передачи, хранения или обработки информации. Кодирование не меняет сообщение, не затрудняет его чтение, наоборот облегчает обработку сообщения (перевод сообщения из русского текста в азбуку морзе или телеграфный код).

**Шифрование** — подготовка сообщения таким образом, чтобы промежуточный нелегальный участник не смог его прочитать.

**Открытый текст** — текст до преобразования.

**Шифрованный текст (шифротекст)** — текст после шифрования.

**Шифрограмма** — зашифрованный текст пригодный для передачи по каналу связи.

**Функция шифрования** — функция преобразующая открытый текст в шифрованный текст.

**Ключ шифрования** — дополнительный аргумент функции (используя разные ключи получаем разный шифротекст на выходе).

**Ключ расшифрования** — ключ, который должен использовать легальный пользователь, для того, чтобы восстановить текст.

**Кольцо** — это множество с двумя бинарными операциями: сложением и умножением.

**Поле** — это коммутативное аддитивное кольцо с единицей.

**Конечное поле или поле Галуа** — это поле с конечным числом элементов.

**Генератор мультипликативной группы** — это элемент, возводя который во все степени можно получить все элементы группы.

**Классическая криптография (с секретным ключом)** - это симметричная криптосистема, в которой шифруем и расшифровываем одним ключом (𝑍1 = 𝑍2 ). Или если легко получить один из другого.

**Криптография с открытым ключом (на открытых ключах):** Ассиметричная криптосистема — это такая криптосистема, в которой ключ шифрования и расшифрования связаны друг с другом сложными математическими операциями

**Генератор случайных чисел** — это алгоритм (или физический процесс), который воспроизводит случайные числа. В настоящее время для целей криптографии пока что нет дешёвого, быстрого и качественного генератора. Поэтому приходится использовать так называемые **псевдослучайные генераторы (генераторы псевдослучайной последовательности).** Они отличаются от генератора случайной последовательности тем, что являются алгоритмом, то есть можно записать математический закон получения следующего случайного числа на основании предыдущего случайного числа и некоторого состояния вычислительного комплекса.

**Примеры алгоритмов генерации псевдопростых чисел**:

* Решето Эратосфена
* Алгоритм Ферма
* Алгоритме Миллера
* Алгоритм Миллера – Рабина
* Тест AKS

**Хэширование или хеширование** — преобразование массива входных данных произвольной длины в (выходную) битовую строку фиксированной длины, выполняемое определённым алгоритмом. Функция, реализующая алгоритм и выполняющая преобразование, называется «**хеш-функцией**» или «функцией свёртки». Исходные данные называются входным массивом, «ключом» или «сообщением». Результат преобразования (выходные данные) называется «**хешем**», «**хеш-кодом**», «**хеш-суммой**», «сводкой сообщения».

**Хэш-таблица** — это структура данных, реализующая интерфейс ассоциативного массива, а именно, она позволяет хранить пары (ключ, значение) и выполнять три операции: операцию добавления новой пары, операцию поиска и операцию удаления пары по ключу.

**Электронная подпись** — реквизит электронного документа, полученный в результате криптографического преобразования информации с использованием закрытого ключа подписи и позволяющий проверить отсутствие искажения информации в электронном документе с момента формирования подписи (целостность), принадлежность подписи владельцу сертификата ключа подписи (авторство), а в случае успешной проверки подтвердить факт подписания электронного документа (неотказуемость).

**Криптографическая система с открытым ключом** (разновидность асимметричного шифрования, асимметричного шифра) — система шифрования и/или электронной подписи (ЭП), при которой открытый ключ передаётся по открытому (то есть незащищённому, доступному для наблюдения) каналу и используется для проверки ЭП и для шифрования сообщения. Для генерации ЭП и для расшифровки сообщения используется закрытый ключ[1]. (Примеры: Алгоритм Диффи – Хеллмана, Алгоритм RSA)

**Схема Эль-Гамаля (Elgamal)** — криптосистема с открытым ключом, основанная на трудности вычисления дискретных логарифмов в конечном поле. Криптосистема включает в себя алгоритм шифрования и алгоритм цифровой подписи. Схема Эль-Гамаля лежит в основе бывших стандартов электронной цифровой подписи в США (DSA) и России (ГОСТ Р 34.10-94).

* 1. **Ссылки**

**Спецификация требований программного обеспечения IEEE STD 830-1998:**

<http://standards.ieee.org/findstds/standard/830-1998.html>

**Примеры реализованных крипто-библиотек:**

<https://github.com/apache/shiro>

<https://github.com/openssl/openssl>

**Участники проекта:**

Канзепаров Руслан (реализация кода) – <https://github.com/kanzeparov>

Хвостюк Сергей (реализация кода) – <https://github.com/Kamitonishe>

Беренков Никита (юнит тесты и валидация алгоритмов, тестирование на производительность) – <https://github.com/Nikita19960809>

Пушилин Вадим (документация, тестирование на производительность) – <https://github.com/vadpush>

**Ссылка на репозиторий проекта**:

<https://github.com/kanzeparov/CyberSecurity>

* 1. **Краткий обзор**

В нашей библиотеке будут реализованы следующие алгоритмы:

1. Deffie-Hellman
2. ElGamal
3. ГОСТ(Кузнечик) Р 34.12 - 2015
4. PKCS 15

В проекте будут использованы следующие библиотеки:

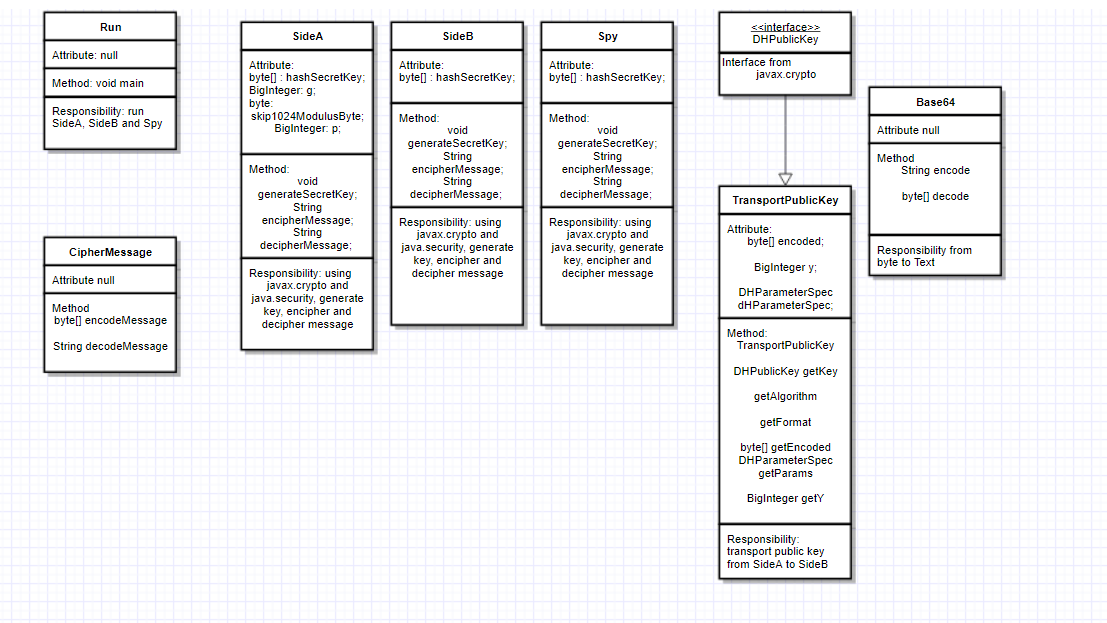
* java.util.crypto
* java.util.security
* apache.shiro (github.com/apache/shiro)

Основным интерфейсом будет реализация классов, пронаследованных от DefaultBlockCipherService с имплементом CipherService.

**Детальные требования**

Требования к внешним интерфейсам

* **Интерфейсы пользователя**



* **Интерфейсы аппаратного обеспечения**

