Защита

**Слайд 1. Тема** моей выпускной квалификационной работы «Исследование методов дискретного логарифмирования».

**Слайд 2. Актуальность** исследования обуславливается тем, что задача дискретного логарифмирования является одной из основных задач, на которых базируется криптография с открытым ключом. Классическими криптографическими схемами на её основе являются схема выработки общего ключа Диффи-Хеллмана, схема электронной подписи Эль-Гамаля, криптосистема Мэсси-Омуры для передачи сообщений.

**Слайд 3. Цель данной работы** – исследование и разработка алгоритмов дискретного логарифмирования с экспоненциальной и субэкспоненциальной сложностью.

Для достижения цели был поставлен ряд **задач**, из которых можно выделить следующие:

- реализовать вспомогательные математические функции для проверки алгоритмов дискретного логарифмирования,

- исследовать и реализовать базовые и модифицированные алгоритмы дискретного логарифмирования,

- провести эксперименты и сравнительный анализ на реализованных базовых и модифицированных методах дискретного логарифмирования.

**Слайд 4. Используемыми технологиями** являются C# на .NET8 в Windows Forms.

**Слайд 5.** Для исследования рабочим **определением** послужило определение дискретного логарифмирования~~, которое звучит следующим образом:~~ ~~«Дискретное логарифмирование является задачей обращения функции в некоторой конечной мультипликативной группе ». Пусть в некоторой конечной мультипликативной абелевой группе задано уравнение:~~

Решение задачи дискретного логарифмирования состоит в нахождении некоторого целого неотрицательного числа , удовлетворяющего уравнению.

**Слайд 6.** В данной работе исследованы алгоритмы экспоненциальной сложности, которые включают в себя алгоритмы Шенкса, Полига-Хеллмана, ро-метод Полларда, а также алгоритмы субэкспоненциальной сложности, а именно алгоритмы Адлемана, COS и решето числового поля.

**Слайд 7. Алгоритм Шенкса.** Была реализована модификация алгоритма, состоящая в распараллеливании алгоритма. При вычислении двух рядов чисел в начале алгоритма теоретическая оценка сложности базового алгоритма , а модифицированного . Далее в данных рядах ищется число, которое есть в обоих рядах асинхронно с начала и конца ряда. Теоретическая оценка сложности данного шага базового алгоритма , а модифицированного .

Модификация лучше во всех тестах.

**Слайд 8. Алгоритм Полига-Хеллмана**. Была реализована модификация алгоритма, состоящая в том, что в начале алгоритма число было разложено на простые множители и данные простые множители были возведены в свои степени, чтобы далее были вычисления с единичными значениями без степеней. Теоретическая оценка сложности вычисления таблицы базового алгоритма , а модифицированного .

Модификация лучше во всех тестах.

**Слайд 9. Ро-метод Полларда.** Была реализована модификация алгоритма, состоящая в том, что в алгоритме увеличилась степень вычисляемого полинома . При вычислении степень полинома увеличилась до 3. Теоретическая оценка сложности базового алгоритма , а модифицированного .

Модификация лучше в скорости при 64 и 256 битах.

**Слайд 10. Алгоритм Адлемана.** Была реализована модификация алгоритма, состоящая в том, что в начале алгоритма был изменён показатель степени при вычислении числа , тем самым понизив факторную базу. Теоретическая оценка сложности базового алгоритма , а модифицированного .

Модифицированный показал себя лучше в тестах в затраченной памяти и в скорости при 32 и 8.

**Слайд 11. Алгоритм COS.** Была реализована модификация алгоритма, состоящая в том, что уравнение было разложено на простые множители и данные простые множители были возведены в свои степени, чтобы далее были вычисления с единичными значениями без степеней. Теоретическая оценка сложности вычисления таблицы базового алгоритма , а модифицированного .

Модифицированный показал себя во всех тестах.

**Слайд 12. Решето числового поля.** Была реализована модификация алгоритма, состоящая в том, что в начале алгоритма выбирается степень неприводимого многочлена, равное количество байт входного числа . (). Теоретическая оценка сложности дальнейшего разложения числа по основанию базового алгоритма , а модифицированного .

Модифицированный лучше во всех тестах.

**Слайд 13. В результате экспериментов можно сделать вывод о том, что** модифицированные экспоненциальные и субэкспоненциальные алгоритмы при определённых параметрах имеют в большинстве тестов лучше показатели в скорости выполнения и затраченной памяти.

**Слайд 14.** Спасибо за внимание. Я готов ответить на ваши вопросы.