**ОТЧЁТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №4**

**1. Общее описание протокола**

Протокол CRISP — это специализированная схема защищенного взаимодействия, предназначенная для обмена сообщениями в индустриальных и автоматизированных системах управления. Его архитектура ориентирована на обеспечение целостности, конфиденциальности и подлинности сообщений между участниками обмена.

**Основные компоненты протокола:**

* **Инициализация:** включает генерацию ключей на основе исходного мастер-ключа и уникальных идентификаторов устройств.
* **Создание пакета:** данные шифруются, после чего к ним добавляется имитовставка (MAC), рассчитываемая по стандарту OMAC.
* **Обработка:** принимающая сторона проверяет MAC, выполняет дешифровку и сверяет номер последовательности для предотвращения повторов.
* **Смена ключей:** реализуется через детерминированную функцию генерации производных ключей (KDF), что позволяет обновлять ключевой материал без внешней передачи секретов.

**2. Криптографические свойства и принципы построения**

**Используемые алгоритмы:**

* **Шифрование:** симметричный блочный алгоритм ГОСТ Магма (256-битный ключ, 32 раунда), работающий в режиме ECB с добавлением паддинга по ГОСТ Р 34.13.
* **Аутентификация:** OMAC (одноразовый MAC), основанный на том же блочном шифре.

**Принципы:**

* **Размер блока:** 64 бита.
* **Ключевая длина:** 256 бит.
* **Обновление ключей:** базируется на функции расширения ключа с использованием меток, счётчиков и соли.
* **Структура сеанса:** каждый сеанс сопровождается уникальной генерацией ключей шифрования и MAC, что препятствует повторному использованию одного и того же ключевого материала.

**3. Вычислительная сложность**

**Временная:**

* Все операции работают с линейной сложностью по объему данных:
  + **Шифрование/дешифрование:** O(n)
  + **Вычисление MAC:** O(n)

**Память:**

* **Контекстное состояние протокола:** около 500 байт (включая ключи, счётчики, идентификаторы).
* **Рабочий буфер для сообщений:** ограничен 2048 байтами в текущей реализации.
* **Промежуточные буферы:** используются только на время одного сообщения, после чего очищаются.

**4. Обоснование выбора протокола**

Протокол CRISP реализует требования отечественного стандарта, что делает его предпочтительным для российских предприятий и объектов критической инфраструктуры.

**Ключевые причины выбора:**

* **ГОСТ-совместимость:** используется утвержденный шифр Магма, который входит в перечень российских криптоалгоритмов.
* **Безопасность:** проверенная схема обмена с аутентификацией и шифрованием, устойчивая к большинству известных атак.
* **Лёгкость реализации:** понятная структура протокола позволяет гибко его адаптировать под специфические задачи.

**5. Программная реализация**

**Организация проекта:**

Код структурирован по функциональным модулям:

* magma\_calc.{c,h} — реализация основного шифра ГОСТ Магма.
* modes.{c,h} — реализация режимов шифрования, включая ECB.
* omac.{c,h} — модуль генерации имитовставки (MAC).
* keygen.{c,h} — логика KDF и управление ключами.
* crisp.{c,h} — реализация логики обмена, включая создание и валидацию сообщений.

**Особенности:**

* Буферизация: все данные обрабатываются в оперативной памяти без временных файлов.
* Паддинг: реализован строго в соответствии с ГОСТ — добавляется байт 0x80, далее нули.
* Очистка: все чувствительные данные стираются после использования.

**6. Отчёт о скорости выполнения тестов**

**6.1 Аутентификация (10⁴ – 10⁶ операций)**

Тест скорости аутентификации

Количество итераций: 452702

Общее время выполнения: 30.30 сек

Среднее время на одну операцию: 66.924 микросекунды

**6.2 Сеансный обмен с обновлением ключей (10² – 10³ итераций)**

Тест скорости сеансов

Количество сеансов: 388

Сообщений на сеанс: 10

Общее время выполнения: 1.01 сек

Среднее время на сообщение: 259.798 микросекунды

**6.3 Загрузка процессора и памяти**

Процент использования CPU: 99%

Общее время выполнения: 0:31.30

Максимальное потребление памяти: 2 352 КБ

**7. Проверка встречной работы**

Проверка встречной совместимости была выполнена следующим образом:

* Одна сторона (инициатор) формирует сообщение с помощью функций crisp\_create\_message.
* Вторая сторона (приемник) обрабатывает сообщение через crisp\_process\_message.
* При идентичных начальных ключах, идентификаторах и счетчиках сообщение расшифровывается корректно, а MAC успешно верифицируется.

Результат: протокол обеспечивает полную симметричность и корректную встречную обработку сообщений в обоих направлениях.