Спецификация по СКЗИ

Раздел "Документация"

|  |  |  |
| --- | --- | --- |
| Код документа | Наименование и файл | Содержание / Назначение |
| СКЗИ-ТП-001 | Текст программы (файл: main-4.c) | Основной исполняемый код системы, тест KDF\_TREE\_GOSTR3411\_2012\_256 |
| СКЗИ-ОП-002 | Описание программы | Реализация криптографических операций и аутентификации. Алгоритмы: KDF\_TREE\_GOSTR3411\_2012\_256, простейшая аутентификация |
| СКЗИ-РП-003 | Руководство пользователя | Инструкция по работе с CLI, описание запуска и параметров тестирования |
| СКЗИ-ЛОГ-004 | Формат журнала аудита (crypto\_audit.log) | Запись событий аутентификации и критических операций |

Раздел "Комплексы"

|  |  |  |
| --- | --- | --- |
| Код комплекса | Наименование комплекса | Краткое описание |
| СКЗИ-КОМПЛЕКС-001 | Криптографический модуль обработки ключей | Включает реализацию KDF\_TREE\_GOSTR3411\_2012\_256 и вспомогательные функции для тестирования производных ключей |

Раздел "Компоненты"

|  |  |  |
| --- | --- | --- |
| Код компонента | Наименование компонента | Краткое описание |
| СКЗИ-МОДУЛЬ-001 | Модуль аутентификации (authenticate\_user) | Проверка пароля, блокировка после 5 неудачных попыток |
| СКЗИ-МОДУЛЬ-002 | Модуль логирования (log\_event) | Запись событий в crypto\_audit.log |
| СКЗИ-МОДУЛЬ-003 | Модуль генерации случайных байтов (generate\_random\_bytes) | Генерация случайных данных для ключей |
| СКЗИ-МОДУЛЬ-004 | Модуль безопасного обнуления памяти (secure\_zero) | Очистка секретных данных из памяти |
| СКЗИ-МОДУЛЬ-005 | Модуль тестирования KDF (main, вызовы KDF\_TREE\_GOSTR3411\_2012\_256) | Проведение тестов на производных ключах и вывод результатов |

Примечания

1. В разделе "Документация" коды видов документов указаны в соответствии с внутренней классификацией.
2. В компонентах используются заголовочные файлы: kdf\_tree.h, perf\_metrics.h, стандартные заголовки C.
3. Реализация соответствует требованиям ГОСТ Р 34.11-2012 (KDF\_TREE\_GOSTR3411\_2012\_256).
4. Для защиты от перебора пароля предусмотрена блокировка после 5 неудачных попыток.

описание ПО СКЗИ

Основные сведения о составе ПО СКЗИ

Состав программного обеспечения включает следующие файлы:

kdf\_tree.h (SHA-256: пример: 2cf24dba5fb0a30e26e83b2ac5b9e29e1b161e5c1fa7425e73043362938b9824) — заголовочный файл с описанием интерфейса KDF\_TREE\_GOSTR3411\_2012\_256.

perf\_metrics.h (SHA-256: пример: 486ea46224d1bb4fb680f34f7c9ad96a8f24ec88be73ea8e5a6c65260e9cb8a7) — заголовочный файл для замеров производительности.

main.c (SHA-256: пример: 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d) — основной исполняемый код системы, реализующий аутентификацию, тестирование и демонстрацию работы KDF.

Логическая структура ПО СКЗИ

1. Модули и функции:
   1. Аутентификация: Проверка пароля пользователя (authenticate\_user), блокировка при превышении 5 попыток (MAX\_FAILED\_AUTH).
2. Логирование:
   1. Запись событий в файл crypto\_audit.log (log\_event).
   2. Генерация случайных данных:
   3. Функция generate\_random\_bytes для получения случайных байтов (используется для генерации ключей)
3. KDF (функция формирования ключей):
   1. Вызов KDF\_TREE\_GOSTR3411\_2012\_256 для получения производных ключей на основе мастер-ключа, метки и соли.
4. Безопасное обнуление памяти:
   1. Функция secure\_zero для очистки конфиденциальных данных
   2. Основной модуль (main):
   3. Координирует аутентификацию, логирование, тестирование KDF, очистку данных

Описание ПО СФ СКЗИ

Специальные функции:

1. Аутентификация:
   1. Статическая проверка пароля "securepass". После 5 неудачных попыток — блокировка на 10 секунд.
2. Формирование ключей:
   1. Использование функции KDF\_TREE\_GOSTR3411\_2012\_256 с поддержкой метки, соли и итераций для генерации производных ключей.
3. Логирование:
   1. Фиксация всех критических событий — успешные и неуспешные попытки входа, завершение работы, ошибки.
4. Защита данных:
   1. Безопасное обнуление мастер-ключа, метки, соли и производных ключей после завершения работы

Методы и правила эксплуатации

1. Требования к запуску:
   1. Программа запускается через командную строку без параметров: ./main
   2. После запуска требуется ввести пароль ("securepass").
2. Ограничения:
   1. После 5 неудачных попыток ввода пароля доступ блокируется на 10 секунд.
   2. Для корректной работы требуется поддержка стандартных функций C и наличие файла журнала (crypto\_audit.log).
3. Рекомендации:
   1. Использовать ОС, поддерживающую функции secure\_zero и стандартные файловые операции.

Инструкция по сборке

1. Скомпилировать исходные файлы командой:

gcc -o main-4 main-4.c kdf\_tree.c perf\_metrics.c -O2 -Wall

1. Проверить контрольные суммы файлов.
2. Запустить исполняемый файл: ./main-4

Описание применения ПО СКЗИ

Назначение ПО СКЗИ

Программное обеспечение СКЗИ реализует криптографическую защиту информации на уровне формирования производных ключей с помощью алгоритма KDF\_TREE\_GOSTR3411\_2012\_256 (деревовидная функция формирования ключа на базе ГОСТ Р 34.11-2012). Основное назначение — обеспечение безопасного и стандартизованного способа получения производных ключей для последующего использования в криптографических протоколах и приложениях.

ПО также реализует базовую аутентификацию пользователя с журналированием событий для контроля доступа к криптографическим операциям и фиксации критических событий.

Область применения ПО СКЗИ

ПО предназначено для использования в информационных системах, где требуется безопасное формирование производных ключей из мастер-ключей, например:

1. генерация ключей для сессионного шифрования;
2. защита ключевого материала при хранении и передаче;
3. обеспечение многоуровневой/иерархической криптографической защиты.

ПО ориентировано на применение в коммерческих, корпоративных и исследовательских системах, не связанных с государственной тайной.

Класс решаемых ПО СКЗИ задач

Программное обеспечение предназначено для решения следующих задач:

1. формирование производных ключей по алгоритму KDF\_TREE\_GOSTR3411\_2012\_256 (ГОСТ Р 34.11-2012);
2. контроль доступа к криптографическим операциям на основе аутентификации пользователя;
3. ведение аудита (журналирования) всех событий, связанных с криптографическими операциями и доступом;
4. безопасное удаление (очистка) секретных данных из памяти после использования.

Ограничения при применении ПО СКЗИ

1. ПО не предназначено для защиты информации, составляющей государственную тайну.
2. Аутентификация реализована в виде заглушки и не должна использоваться в продуктивной среде без замены на полноценный механизм.
3. Поддерживается только работа с фиксированными или сгенерированными в памяти ключами и метками; работа с файлами не реализована.
4. Не реализована защита от атак по сторонним каналам (например, по времени выполнения).
5. В текущей реализации не выполняется постоянное отслеживание повторного использования меток или мастер-ключей.

**Минимальная конфигурация технических средств**

1. Архитектура: x86 или ARM (32/64-бит);
2. Оперативная память: не менее 32 МБ;
3. Процессор: не ниже Intel Pentium III / ARM Cortex-A7;
4. Операционная система: POSIX-совместимая (Linux, FreeBSD), bash или sh-оболочка;
5. Доступ к файловой системе (для создания журнала аудита);
6. Поддержка стандартной библиотеки C.

Сведения о среде функционирования (СФ)

Среда функционирования включает:

1. POSIX-совместимую ОС (Linux);
2. стандартную библиотеку языка C (libc);
3. подключаемые модули (kdf\_tree.h, perf\_metrics.h);
4. файловую систему с поддержкой прав доступа;
5. потоковый ввод-вывод (системные вызовы open, read, write).

Порядок работы ПО СКЗИ

1. Пользователь запускает ПО из командной строки.
2. При необходимости выполняется проверка времени блокировки (если ранее было превышено количество неуспешных аутентификаций).
3. Пользователь проходит процедуру аутентификации (ввод пароля "securepass").
4. Выполняется тест формирования производного ключа на основе фиксированного мастер-ключа, метки и соли.
5. Производится вывод результатов (мастер-ключ и производный ключ).
6. Все критические события фиксируются в журнале аудита.
7. Все секретные данные (ключи, метки, соль) очищаются из памяти.
8. Программа завершает выполнение.

Пояснительная записка

Назначение компонентов, входящих в состав ПО СКЗИ

Программное обеспечение состоит из следующих компонентов.  
Модуль main-4.c управляет выполнением программы, реализует пользовательский интерфейс, аутентификацию, вызов криптографической функции KDF\_TREE\_GOSTR3411\_2012\_256, вывод результатов, очистку секретных данных и ведение журнала событий.  
Модуль kdf\_tree.h реализует деревовидную функцию формирования производных ключей по ГОСТ Р 34.11-2012 (KDF\_TREE\_GOSTR3411\_2012\_256).  
Модуль perf\_metrics.h (и соответствующая реализация) предназначен для замеров производительности при массовом формировании производных ключей (используется в тестах).  
Встроенные функции main-4.c обеспечивают: генерацию случайных байтов (generate\_random\_bytes), безопасное обнуление памяти (secure\_zero), аутентификацию пользователя (authenticate\_user), журналирование событий (log\_event).

Перечень всех реализованных в ПО СКЗИ функций

1. main — основная управляющая функция, координирующая выполнение программы, тестирование KDF, очистку данных.
2. authenticate\_user — функция аутентификации пользователя.
3. log\_event — функция записи событий в журнал аудита.
4. secure\_zero — функция безопасного обнуления памяти.
5. generate\_random\_bytes — функция генерации случайных байтов для ключей.
6. KDF\_TREE\_GOSTR3411\_2012\_256 — криптографическая функция формирования производных ключей (экспортируется из kdf\_tree.h).
7. print\_hex — вспомогательная функция для вывода ключей в шестнадцатеричном виде.

Сведения о параметрах реализованных функций

1. authenticate\_user не принимает аргументов, возвращает логическое значение (1 — успешная аутентификация, 0 — неудача).
2. log\_event принимает строку с сообщением и записывает её в файл журнала.
3. secure\_zero принимает указатель на память и размер области, очищает её.
4. generate\_random\_bytes принимает буфер и его размер, заполняет буфер случайными байтами.
5. KDF\_TREE\_GOSTR3411\_2012\_256 принимает мастер-ключ, длину ключа, метку, соль, номер итерации, длину производного ключа и выходной буфер.
6. print\_hex принимает описание, буфер и его длину, выводит содержимое буфера в шестнадцатеричном виде.

Сведения о формируемых кодах возврата реализованных функций

1. main возвращает код 0 при успешном завершении работы, в случае ошибок (например, превышено число попыток аутентификации, сбой генерации ключа) — завершает выполнение с помощью exit(1).
2. authenticate\_user возвращает 1 при успешной аутентификации, 0 при ошибке.
3. В случае критических ошибок (например, сбой открытия файла журнала) также вызывается exit(1).

Перечень экспортируемых функций ПО СКЗИ

Из модуля kdf\_tree.h экспортируется функция KDF\_TREE\_GOSTR3411\_2012\_256 для формирования производных ключей.

Из модуля perf\_metrics.h экспортируются функции для замеров производительности (используются при тестировании).

Вспомогательные функции (generate\_random\_bytes, secure\_zero, log\_event) используются внутри main-4.c.

**Описание используемых переменных**

1. master\_key — массив из 32 байт (256 бит), используется как мастер-ключ для KDF.
2. label — строка (метка) для KDF, задаёт контекст производного ключа.
3. seed — строка (соль) для KDF, повышает криптостойкость производного ключа.
4. derived\_key — буфер для полученного производного ключа (размер задаётся переменной L).
5. failed\_attempts — количество неудачных попыток аутентификации.
6. blocked\_until — время, до которого заблокирован повторный ввод пароля.
7. password — временный буфер для ввода пароля пользователем.

Описание алгоритмов функционирования ПО СКЗИ

После запуска программа проверяет, истекло ли время блокировки пользователя. Если блокировка не активна, пользователь проходит процедуру аутентификации. При успешной аутентификации выполняется тест формирования производного ключа на основе фиксированного мастер-ключа, метки и соли с помощью функции KDF\_TREE\_GOSTR3411\_2012\_256. Результаты выводятся в консоль в шестнадцатеричном виде. Все события (успешная или неуспешная аутентификация, завершение работы) фиксируются в журнале аудита. После завершения работы все секретные данные (мастер-ключ, метка, соль, derived\_key) очищаются из памяти функцией secure\_zero. Программа завершает выполнение.

Описание критериев, методики и результатов тестирования

Критерии тестирования включали: корректность формирования производного ключа (сравнение с эталонным значением), реакцию на неверный пароль, ведение журнала событий, очистку памяти. Методика тестирования включала ручной запуск программы, ввод корректного и некорректного пароля, анализ содержимого журнала аудита, проверку очистки секретных данных. В результате тестирования установлено, что при корректных входных данных функция KDF\_TREE\_GOSTR3411\_2012\_256 формирует производный ключ, совпадающий с эталонным, при ошибках аутентификации программа блокирует доступ, а журнал фиксирует все ключевые события. Все функции отработали в соответствии с требованиями.