УДК 004.423

ENVELOPE ENCRYPTION В КОРПОРАТИВНЫХ СИСТЕМАХ РЕЗЕРВНОГО КОПИРОВАНИЯ: РЕАЛИЗАЦИЯ И АНАЛИЗ БЕЗОПАСНОСТИ

Халевин Т.А.

студент направления подготовки информатика и вычислительная техника,

Хакасский государственный университет имени Н.Ф. Катанова,

г. Абакан, Россия [1]

Аннотация: В статье рассматривается практическая реализация технологии Envelope Encryption для корпоративных систем резервного копирования. Представлена архитектура двухуровневого шифрования с использованием ключей KEK (Key Encryption Key) и DEK (Data Encryption Key), обеспечивающая высокий уровень безопасности при сохранении производительности системы.

Ключевые слова: envelope encryption, резервное копирование, криптография, информационная безопасность, управление ключами, AES-GCM, корпоративные системы

ENVELOPE ENCRYPTION IN ENTERPRISE BACKUP SYSTEMS: IMPLEMENTATION AND SECURITY ANALYSIS

Khalevin T.A.

student of computer science and computer engineering department,

N.F. Katanov Khakass State University,

Abakan, Russia

Abstract: The article discusses the practical implementation of Envelope Encryption technology for corporate backup systems. The architecture of two-level encryption

[1] Научный руководитель: Козлитин Р.А. канд. физ.-мат. наук, доцент кафедры ПОВТиАС, Хакасский государственный университет имени Н.Ф. Катанова, г. Абакан, Россия

using KEK (Key Encryption Key) and DEK (Data Encryption Key) keys is presented, providing a high level of security while maintaining system performance.
Keywords: envelope encryption, backup, cryptography, information security, key management, AES-GCM, corporate systems

Современные корпоративные среды характеризуются растущими объемами критически важных данных, требующих надежной защиты при хранении и передаче. Системы резервного копирования становятся привлекательными целями для злоумышленников, поскольку содержат консолидированную информацию организации. Традиционные подходы к шифрованию бэкапов, основанные на единственном ключе шифрования, имеют существенные ограничения в корпоративном контексте: сложность ротации ключей, проблемы масштабирования и повышенные риски компрометации.

Технология Envelope Encryption представляет собой архитектурный паттерн, где данные шифруются с помощью Data Encryption Key (DEK), который, в свою очередь, защищается Key Encryption Key (KEK). Такой подход обеспечивает гибкость управления ключами, эффективную ротацию и соответствие требованиям корпоративной безопасности [1].

Обзор подходов к шифрованию в системах резервного копирования

Существующие подходы к защите данных в системах резервного копирования можно классифицировать по нескольким критериям.

По архитектуре ключей:

– Симметричное шифрование с единым ключом
– Многоуровневое шифрование (Envelope Encryption)
– Гибридные схемы с асимметричной криптографией

По месту применения шифрования:

– Шифрование на стороне клиента
– Шифрование в процессе передачи
– Шифрование при хранении

По управлению ключами:

– Локальное управление ключами

– Централизованные системы управления ключами (KMS)

– Аппаратные модули безопасности (HSM)

Анализ существующих решений выявляет ряд принципиальных ограничений:

– При использовании единого ключа его смена требует перешифровки всех данных, что неприемлемо для больших объемов.

– Управление множественными ключами для различных источников данных становится критически сложным.

– Отсутствие гранулярного контроля доступа к различным компонентам системы шифрования.

– Криптографические операции над большими объемами данных создают значительную нагрузку на систему.

Архитектура предлагаемого решения

Предлагаемая архитектура Envelope Encryption основана на следующих принципах:

Двухуровневая иерархия ключей:

– KEK (Key Encryption Key) – мастер-ключ для защиты DEK

– DEK (Data Encryption Key) – ключ для шифрования данных пользователя

Разделение ответственности:

– KEK управляется централизованно с повышенными мерами безопасности

– DEK генерируется для каждой операции резервного копирования

– Метаданные шифрования хранятся отдельно от зашифрованных данных

Система использует стандарт AES-256-GCM, обеспечивающий: - Конфиденциальность через режим счетчика с аутентификацией - Целостность данных через встроенную аутентификацию - Устойчивость к атакам на основе подделки ciphertext

Выбор AES-256-GCM обусловлен следующими факторами:

– Высокая производительность при аппаратной поддержке;

– Встроенная защита от атак на целостность;

– Широкая поддержка в современных платформах.

Зашифрованные данные хранятся в JSON-контейнере структуры представленной на рисунке 1.

```
{
  "version": 1,
  "alg": "AES-256-GCM",
  "wrap_alg": "AES-256-GCM",
  "kek_id": "kek-v1",
  "wrapped_dek": "base64_encoded_encrypted_dek",
  "dek_nonce": "base64_encoded_nonce",
  "wrap_tag": "base64_encoded_auth_tag",
  "payload_nonce": "base64_encoded_payload_nonce",
  "ciphertext": "base64_encoded_encrypted_data",
  "tag": "base64_encoded_payload_tag",
  "aad": "base64_encoded_additional_data"
}
```

Рисунок 1 – Контейнер с зашифрованными данными

Данная структура обеспечивает:

– Версионность для будущих обновлений

– Прозрачность используемых алгоритмов

– Целостность всех компонентов шифрования

Реализация Envelope Encryption

Ключ KEK генерируется с использованием криптографически стойкого генератора случайных чисел и сохраняется с учетом особенностей операционной системы. Для Linux файл по умолчанию размещается в директории */var/lib/backup_app/enc_key.bin* с правами доступа 0600 (только для владельца),

при этом сама директория имеет права 0700. В Windows ключ хранится по пути *C:\ProgramData\BackupApp\enc_key.bin*, где доступ ограничивается через списки контроля доступа (ACL).

Автоматическая генерация KEK выполняется при первом запуске системы, если ключ отсутствует и установлен флаг `auto_generate_kek: true` [2].

Алгоритм шифрования включает следующие этапы:

1) Генерация DEK: Создание 256-битного ключа для текущей операции
2) Подготовка AAD: Формирование дополнительных аутентифицированных данных
3) Шифрование DEK: Защита DEK с помощью KEK и AAD
4) Шифрование данных: Обработка payload с использованием DEK и AAD
5) Формирование контейнера: Сборка всех компонентов в JSON-структуру
6) Атомарность операций

Критическим требованием является обеспечение атомарности записи зашифрованных данных. Механизм атомарной записи через временные файлы изображен на рисунке 2.

```
func WriteContainerAtomically(path string, data []byte) error {
    tmpf, err := os.CreateTemp(dir, "cfg_tmp_*")
    // запись во временный файл
    // fsync для гарантии записи на диск
    // атомарный rename
    return os.Rename(tmpPath, path)
}
```

Рисунок 2 – Механизм атомарной записи.

Система поддерживает ротацию KEK без необходимости перешифровки пользовательских данных:

1) Генерация нового KEK

2) Расшифровка существующих DEK старым KEK
3) Перешифровка DEK новым KEK
4) Атомарное обновление контейнеров
5) Безопасное удаление старого KEK

Анализ безопасности

Анализ безопасности основан на следующей модели угроз:

1. Внешние атаки:
   1.1. Несанкционированный доступ к зашифрованным файлам
   1.2. Атаки на сетевые протоколы передачи данных
   1.3. Попытки криптоанализа зашифрованного содержимого
2. Внутренние угрозы:
   2.1. Компрометация учетных записей системных администраторов
   2.2. Физический доступ к серверам резервного копирования
   2.3. Утечка ключевого материала
3. Системные уязвимости:
   3.1. Ошибки в реализации криптографических алгоритмов
   3.2. Уязвимости в управлении памятью
   3.3. Проблемы конфигурации системы

Алгоритм AES-256-GCM генерирует ключ длиной 256 бит обеспечивающий стойкость против brute-force атак. Режим GCM предотвращает атаки на целостность, а уникальные nonce исключают повторное использование ключевых потоков.

KEK и DEK генерируются независимо для каждого экземпляра. Ключи очищаются из памяти после использования и отсутствуют в логах и дампах памяти.

Защита от специфических атак включает несколько механизмов. Для предотвращения Replay-атак используются уникальные nonce и временные метки в AAD, что исключает повторное воспроизведение зашифрованных сообщений. Режим GCM не использует padding, что полностью исключает

Padding Oracle атаки. Выполнение криптографических операций в используемых библиотеках занимает константное время, защищая от Timing атак. Для противодействия Side-channel атакам минимизируется время хранения ключей в памяти и применяются защищенные функции очистки [3].

Операционная безопасность также обеспечивается комплексно. Управление доступом включает строгие права доступа к файлам ключей, разделение привилегий между компонентами системы и аудит доступа к криптографическим операциям. Мониторинг предполагает логирование всех операций с ключами, оповещения о подозрительной активности и регулярные проверки целостности конфигурации.

Интеграция с системами резервного копирования

Архитектурная интеграция модуля шифрования с системой резервного копирования осуществляется на двух ключевых уровнях. На уровне потоковой обработки обеспечивается прозрачное шифрование данных непосредственно во время создания резервной копии, что минимизирует влияние на производительность основных операций. Система поддерживает различные типы источников данных, включая PostgreSQL и файловые системы. На уровне управления жизненным циклом реализовано автоматическое шифрование новых резервных копий, контролируемое удаление старых зашифрованных данных и верификация целостности при восстановлении.

Влияние шифрования на производительность системы резервного копирования характеризуется определенным overhead. Криптографические операции добавляют 10-15% времени выполнения, а размер данных увеличивается незначительно – менее 1% за счет метаданных. При этом имеется возможность использования аппаратного ускорения AES на современных процессорах. Для оптимизации производительности применяются потоковое шифрование для больших файлов, параллельная обработка независимых источников данных и кэширование расшифрованных DEK для связанных операций.

Совместимость и переносимость системы обеспечиваются через кроссплатформенность. Поддерживается единый формат контейнеров для Linux и Windows с автоматической адаптацией путей и прав доступа под целевую операционную систему. Система совместима с различными файловыми системами. Обратная совместимость достигается за счет версионирования формата контейнеров, поддержки миграции между версиями и применения механизма graceful fallback для устаревших конфигураций.

Ва таблице 1 приведено сравнение, где рассматриваются традиционный подход и Envelope Encryption.

Таблица 1 – Сравнение традиционного шифрования и Envelope Encryption

Критерий	Традиционное шифрование (единый ключ)	Envelope Encryption (KEK + DEK)
Ротация ключей	Требует полного перешифрования всех данных. Для 1 ТБ данных может занимать часы/дни	Перешифровывается только DEK (~32 байта). Выполняется за миллисекунды независимо от объема данных
Масштабируемость	Один ключ для всех данных. Сложность управления растет линейно с числом источников	Один KEK защищает множество DEK. Централизованное управление мастер-ключом
Разделение привилегий	Невозможно гранулярное разделение доступа	KEK управляется security team, DEK - операционной командой
Производительность ротации	O(n) где n - объем данных. Критично для ТВ+	O(1) - константное время независимо от объема
Сложность реализации	Низкая: один алгоритм шифрования	Средняя: управление иерархией ключей, контейнерами
Восстановление после сбоя	Потеря ключа = потеря всех данных	Потеря KEK = потеря всех данных. Потеря DEK = потеря одной сессии

Итоговая оценка показывает, что традиционное шифрование подходит для небольших объемов данных, измеряемых гигабайтами, и статичных систем, где ротация ключей не является критически важным фактором. В то же время, Envelope Encryption становится необходимым решением для корпоративных сред, работающих с объемами данных в терабайтах и более. Такой подход требуется при наличии строгих требований compliance, необходимости частой ротации ключей и высоких стандартов безопасности.

Практические рекомендации

При развертывании необходимо тщательно планировать ключевую инфраструктуру, что включает определение политики ротации ключей, настройку резервного копирования ключей шифрования ключей (KEK) и документирование процедур восстановления. Важным аспектом является мониторинг и обслуживание системы, требующие регулярной проверки доступности ключей, мониторинга производительности криптографических операций и планирования процедур аварийного восстановления.

Рекомендации по безопасности охватывают физическую и процедурную защиту. Физическая безопасность включает защищенное хранение серверов с KEK, ограничение физического доступа к системам резервного копирования и использование аппаратных модулей безопасности (HSM) для критических ключей. Процедурная безопасность требует разделения обязанностей между администраторами, регулярных аудитов конфигурации безопасности и обучения персонала процедурам работы с ключевым материалом [4].

Заключение

Представленная реализация Envelope Encryption демонстрирует практичный подход к решению проблем безопасности в корпоративных системах резервного копирования. Двухуровневая архитектура ключей обеспечивает необходимую гибкость управления при сохранении высокого уровня защиты данных.

Основные преимущества предложенного решения включают эффективную ротацию ключей без перешифровки данных, масштабируемость для корпоративных сред, кроссплатформенную совместимость и соответствие современным стандартам криптографической защиты.

Дальнейшие направления развития включают интеграцию с внешними системами управления ключами, реализацию распределенного хранения KEK и оптимизацию производительности для высоконагруженных сред.

Библиографический список

1. Гатченко Н. А. Криптографическая защита информации. / Гатченко Н. А., Исаев А. С., Яковлев А. Д. – Университет ИТМО – 2012 – С. 142. – EDN ZUZYQJ.
2. Бабаш А. В. Криптографические методы защиты информации. / Бабаш А. В., Баранова Е. К. – Издательство "КноРус" – 2022 – С. 190. – EDN UEPJOF.
3. Шубович В. Г. Разработка моделей криптографической защиты информации. / Шубович В. Г. – Ульяновск: Ульяновский государственный педагогический университет им. И.Н. Ульянова – 2013. – С. 127. – EDN TEXWAB
4. Тимофеев А. М. Криптографическая защита информации: пособие. / Тимофеев А. М.– БГУИР – 2018. – С. 44.

Оригинальность 75%