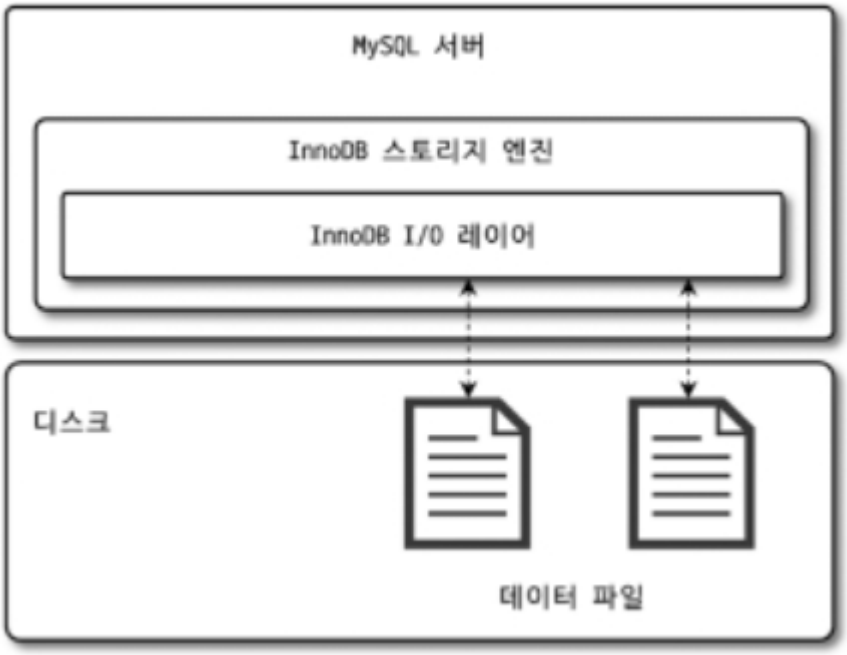
7. 데이터 암호화

↗ Books	Real MySQL 8.0

1. MySQL 서버의 데이터 암호화
    1. 2단계 키 관리
    2. 암호화와 성능
    3. 암호화와 복제

MySQL 5.7 버전부터 지원되기 시작한 데이터 암호화 기능은 처음에는 데이터 파일에 대해서만 제공됐다.
그러다 MySQL 8.0으로 업그레이드되면서 데이터 파일뿐만 아니라 리두 로그나 언두 로그, 바이너리 로그 등도 모두 암호화 기능을 지원하기 시작했다.

1. MySQL 서버의 데이터 암호화



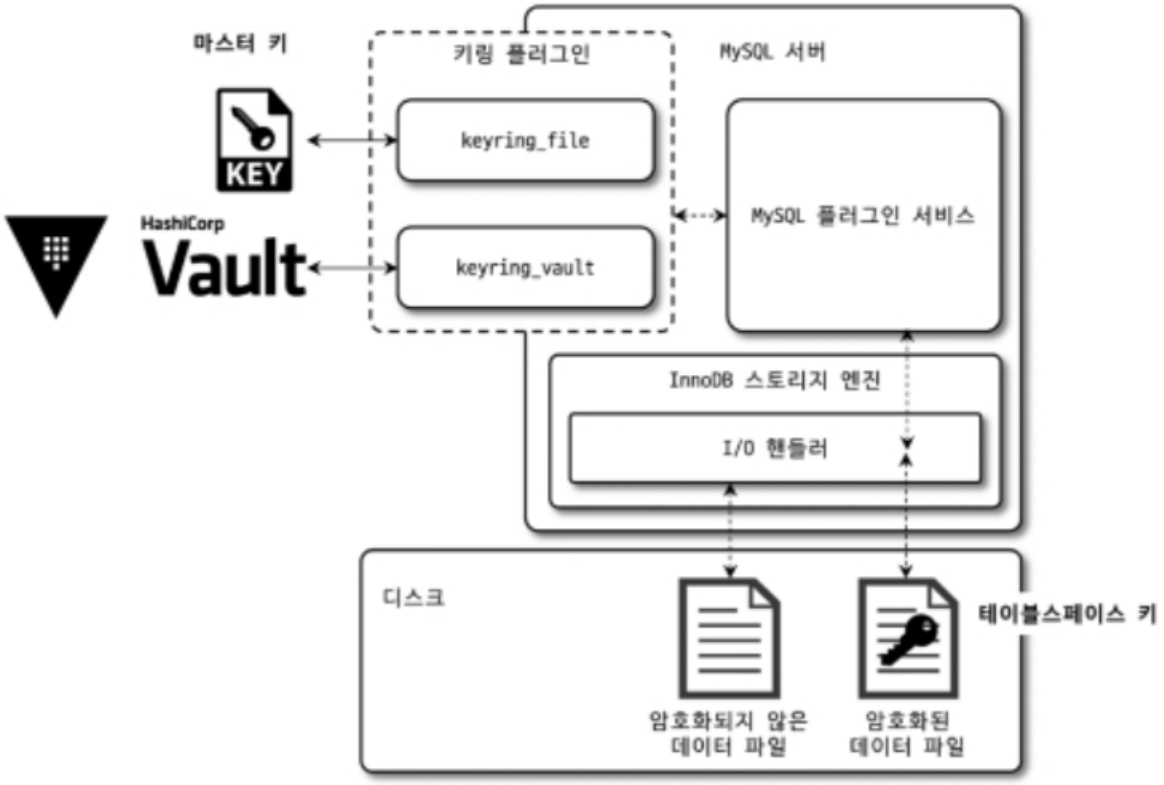
MySQL 서버의 암호화 기능은 그림과 같이 데이터베이스 서버와 디스크 사이의 데이터 읽고 쓰기 지점에서 암호화 또는 복호화를 수행한다. 즉, MySQL 서버에서 디스크 입출력 이외의 부분에서는 암호화 처리가 전혀 필요하지 않고 I/O 레이어에서만 데이터의 암호화 및 복호화 과정이 실행되는 것이다.

이 말은 사용자의 쿼리를 처리하는 과정에서 암호화 여부를 식별할 필요가 없다는 뜻이다. MySQL 내부와 사용자 입장에서는 암호화 활성화 여부에 따른 차이가 없기 때문에 이러한

암호화 방식을 TDE(Transparent Data Encryption)이라고 한다.

1. 2단계 키 관리

MySQL 서버의 TDE에서 암호화 키는 키링(KeyRing) 플러그인에 의해 관리된다. 키링 플러그인은 2단계(2-Tier) 키 관리 방식을 사용한다.



MySQL 서버의 데이터 암호화는 마스터 키(master key)와 테이블스페이스 키(tablespace key)라는 두 종류의 키를 가지고 있다.

MySQL 서버는 외부 키관리 솔루션 또는 디스크의 파일에서 마스터 키를 가져오고, 암호화된 테이블이 생성될 때마다 해당 테이블을 위한 임의의 테이블스페이스 키를 발급한다.

그리고 마스터 키를 이용해 테이블스페이스 키를 암호화해서 각 테이블의 데이터 파일 헤더에 저장한다. 테이블스페이스 키는 절대 MySQL 서버 외부로 노출되지 않기 때문에 주기적으로 변경하지 않아도 보안상 취약점이 되지 않는다.

이렇게 2단계 암호화 방식을 사용하는 이유는 암호화 키 변경으로 인한 과도한 시스템 부하를 피하기 위해서다. 테이블스페이스 키가 변경된다면 모든 데이터를 복호화했다가 변경된

키로 암호화하는 과정을 거쳐야 한다.

> 💡 TDE에서 지원되는 암호화 알고리즘은 AES 256비트다.
> 테이블스페이스 키는 AES-256 ECB 알고리즘을 이용하고 실제 데이터 파일은 AES-256 CBC 알고리즘을 이용해 암호화한다.

2. 암호화와 성능

성능 측면에서 크게 두 가지로 볼 수 있다.

쿼리 처리 성능

버퍼 풀에 존재하지 않는 데이터 페이지를 읽어야 하는 경우 복호화 시간동안 쿼리 처리가 지연될 것이다.
그리고 암호화된 테이블이 변경되면 다시 디스크로 동기화될 때 암호화돼야 하기 때문에 디스크에 저장할 때도 추가로 시간이 더 걸린다.

하지만 데이터 페이지 저장은 백그라운드 스레드가 수행하기 때문에 실제 사용자 쿼리가 지연되는 것은 아니다.

메모리 효율

AES 암호화 알고리즘은 암호화 결과가 평문의 결과와 동일한 크기의 암호문을 반환한다. 따라서 암호화한다고 해서 InnoDB 버퍼 풀의 효율이 달라지거나 메모리 사용 효율이 떨어지는 현상은 발생하지 않는다.

3. 암호화와 복제

레플리카 서버는 소스 서버의 모든 사용자 데이터를 동기화하기 때문에 실제 데이터 파일도 동일할 것이라고 생각할 수 있다. 하지만 마스터 키와 테이블스페이스 키는 서로 다르게 갖도록 설정해야 하기 때문에 암호화 이전 값이 동일하더라도 암호화된 데이터가 저장된 파일의 내용은 완전히 달라진다.