보안 실시간 데이터베이스 시스템에서 시간성 향상을 위한 동적 암호화 관리자에 관한 연구

이순조

A Study on the Dynamic Encryption Manager for Improved Timeliness in Secure Real-Time Database Systems

Soon-Jo Lee*

요 약

대부분의 실시간 응용에서 보안은 또 하나의 중요한 요구 사항이 되고 있다. 따라서 보안 실시간 데이터베이스 시스템은 다양한 보안 등급을 갖는 여러 사용자가 공유하거나 악의를 갖은 해커에 의해 공격될 수 있는 민감한 정보를 관리해야 한다. 보안 실시간 데이터베이스 시스템의 보안을 위해서는 기존의 보안 방법뿐만 아니라 암호화 정책을 적용하는 것이 필요하다. 그렇지만 민감한 정보가 실시간 시스템에서도 보호되어야만 함에도 불구하고 보안 실시간 데이터베이스 시스템에서 암호화 정책을 지원하기 위한 연구가 많지 않았다. 본 논문에서는 보안 실시간 데이터베이스 시스템의 이러한 암호화 정책 문제를 해결하기 위한 암호화 관리자를 제안한다. 보안 실시간 데이터베이스 시스템의 암호화 정책에서 중요한 것은 보안성과 시간성이다. 제안된 암호화 관리자의 중요한 특징은 트랜잭션 데드라인과 보안 수준을 고려하여 데이터를 동적으로 암호화하는 것이다.

ABSTRACT

In many real-time applications, security is another important requirement, since the secure real time database system maintains sensitive information to be shared by multiple users with different levels of security clearance or to be attacked by hackers with ill will. Encryption policies are necessary for the security of secure real-time database systems in addition to the existing security methods, too. However, there has not been much work for the encryption policies in secure real-time database systems, although sensitive information must be safeguarded in real-time systems as well. In this paper, we propose a encryption manager for the purpose of solving the encryption policies of the secure real-time database systems. What is important in the encryption policies of secure real-time database systems is security and timeliness. A significant feature of the proposed encryption manager is the ability to dynamically adapt a encryption algorithm that consider transaction deadline and security level.

키워드

실시간 데이터베이스 시스템, 보안, 암호화, 동적 암호화 관리자

Ⅰ. 서 론

현재까지의 실시간 데이터베이스 시스템에 대한 보안 연구는 일반적인 데이터베이스 시스템에서의 보안 기법을 변형하는 것이었다. 즉, 시간제약 조건과 동시성을 만족하면서 시스템의 최대 성공률

* 서원대학교 컴퓨터정보통신공학부

접수일자 : 2004. 10. 19

을 얻을 수 있는 방법으로 변형된 것이 대부분이었다[1,2]. 그러나 실시간 데이터베이스 시스템에 적용된 변형된 보안 알고리즘도 다음과 같은 이유로 공격자의 공격에 대한 완벽한 방법이라고 할 수는 없다.

대부분의 상용 데이터베이스 시스템들이 채택하고 있는 보안 유지 방법은 데이터에 대한 사용자들의 사용 권한을 제어하는 임의적 접근 제어(DAC : Discretionary Access Control) 방식들이다[3]. 임의적 접근 제어 방식은 보안 유지에 취약점을 드러내고 있는데 이는 데이터에 대한 사용 권한을 사용자 임의대로 다른 사용자들에게 양도할 수 있는 접근 제어 방식이기 때문이다. 이러한 임의적 접근 제어 방식은 대부분의 정직한 내부 사용자들에 대한 정보의 누출을 방지하는 경우에는 적합할 수 있으나, 악의적인 침입자들이 트로이 목마[3, 4]를 이용한 데이터의 접근 또는 컴퓨터 바이러스에 의한 데이터의 접근은 반드시 제한되고 방지되어야 함에도 불구하고 원천적으로 방지할 수 없는 결함을 가지고 있다.

이러한 임의적 접근 제어 방식의 결점을 극복하기 위해 개발된 강제적인 접근 제어(MAC : Mandatory Access Control) 방식은 주체와 객체에 의해 기술된 Bell-LaPadula 모델에 기초한다[5, 6]. 객체는 데이터 파일, 레코드 또는 레코드 내의 필드이고, 주체는 객체들에 대한 접근을 요청할 수 있는 활성화된 프로세스이다. 모든 객체는 비밀 등급이 할당되며, 각각의 주체도 등급별 비밀 취급 인가가 결정되어야 한다. 그렇지만 이 방법도 시스템에 설계되지 않은 비밀 채널을 통하여 상위 보안 등급의 정보가 하위 보안등급으로 흐르는 경우가 발생한다.

이와 같은 이유로 실시간 데이터베이스 시스템에서도 적용된 보안 기법이 공격자에 의해 파괴되었을 경우를 대비하여 중요한 데이터에는 암호화 기법을 적용하는 것이 필요하다. 다만, 기존의 암호화 기법을 그대로 적용하는 것은 실시간 트랜잭션의 처리 율을 저하시키는 요인이 될 것이다. 보안 실시간 데이터베이스 시스템을 구성하고 있는 데이터가 실시간 데이터, 시스템 데이터, 비실시간 데이터로 구분되고 있음에 따라 적용되는 암호화 기법도 각각의 데이터 분류에 맞게 적합하게 적용되어야 한다. 따라서 본 논문에서는 보안 실시간 데이터베이스 시스템에서 발생하는 실시간 트랜잭션의 성공률을 최대화하면서도 보안성을 유지할 수 있는 동적 암호화 관리자를 제시하고자 한다.

본 논문의 2장에서는 관련 연구로서 암호화 기법과 데이터의 형태에 대해 살펴본다. 3장에서는 보안등급과 가중치 부여 방안에 대해 제안한다. 4장에서는 보안 실시간 데이터베이스 시스템에 적용될 동적 암호화 관리자에 대해 제안하고 제안된 기법에 대한 성능 평가를 수행한다. 마지막으로 5장에서는 본 논문에 대한 결론과 향후 연구 방향에 대하여 논한다.

II. 관련 연구

2.1 암호화 기법

암호화 기법이란 평문을 해독 불가능한 형태의 암호문으로 변형하거나 또는 암호문을 해독 가능한 형태로 변환하기 위한 원리 및 방법을 말한다. 일반적인 데이터베이스 시스템에서의 암호화 기법을 도식화하면 아래 그림 2.1과 같다.

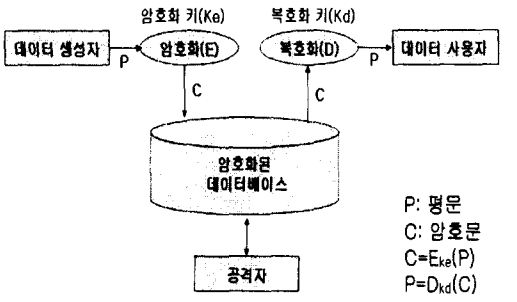


그림 2.4 데이터베이스에서의 암호화 기법
Fig. 2.1 Encryption method of database

데이터 생성자는 평문으로 된 데이터(P)와 암호화 키(Ke)를 암호 알고리즘(E)에 입력하여 암호문(C)을 생성한다. 데이터베이스에 저장된 암호화된 데이터가 공격자에 의해 노출되어도 복호화 키(Kd)를 알 수 없기 때문에 공격자는 평문을 얻기 어렵다.

현재 사용되고 있는 암호화 기법은 암호화와 복호화에 사용되는 키에 의해 대칭키 암호 방식과 공개키 암호 방식으로 나눌 수 있다[7, 8]. 대칭키 암호화 방식은 암호화 키(Ke)와 복호화 키(Kd)가 동일하고 공개키 암호 방식은 두 키가 다르다. 대칭키 방식에서 데이터 생성자와 사용자는 사전에 키를 공유하여야 하지만 공개키와 비교해서 암호화 속도가 빠르고 키가 작다는 장점이 있다. 공개키 방식은 키 분배의 필요성을 없앴으며 디지털 서명과 같은 인증 필요시 쉽게 적용할 수 있는 장점이 있다.

그렇지만 이러한 일반적인 데이터베이스에서의

암호화 기법을 보안 실시간 데이터베이스 시스템에 그대로 적용하기에는 다음과 같은 문제가 있다. 먼저 보안 실시간 데이터베이스의 트랜잭션은 데드라인을 갖고 있기 때문에 적용된 암호화 기법의 암호화 및 복호화에 소요되는 시간이 중요하게 고려되어져야 한다. 또한 실시간 트랜잭션이 하드, 펌, 소프트 트랜잭션 중 어디에 속하는가에 따라 적용되는 암호화 기법도 달라져야한다. 즉, 보안성과 데드라인 내 수행 실패 시에 따른 손실도 중 어디에 중점을 둘 것인가가 중요한 고려요소가 된다. 마지막으로 처리되는 데이터가 실시간 데이터인지 아니면 비실시간 및 시스템 데이터인지도 암호화 기법 적용 시 중요한 고려요소이다. 따라서 보안 실시간 데이터베이스 시스템에 적용될 암호화 기법은 암호화에 적용되는 방법에 따른 보안 수준과 처리 속도에 따라 능동적으로 변형되어야 한다.

2.2 암호화 데이터의 분류

보안 실시간 데이터베이스 시스템에서 보안과 시간 제약 조건이라는 두 가지 제약사항을 만족하기 위해서는 다음과 같이 암호화 대상인 데이터를 분류할 필요가 있다.

보안 실시간 데이터베이스 시스템을 구성하고 있는 데이터에는 실시간 데이터와 비실시간 데이터 그리고 시스템 데이터로 구분할 수 있다[9]. 그림 2.2에서 보는 바와 같이 비실시간 트랜잭션은 모든 데이터를 요구하는데 반해서 실시간 트랜잭션은 실시간 데이터와 시스템 데이터만을 요구한다. 이것이 가능하기 위해서는 데이터베이스 관리자가 데이터베이스 정의 시에 데이터를 실시간과 비실시간으로 정확하게 구분하여야 한다. 이와 같이 보안 실시간 데이터베이스를 구성하고 있는 데이터를 그 형태에 의해 구분함으로써, 데이터 관리자가 실시간 트랜잭션의 데드라인과 중요도에 의해 데이터의 암호화 정책을 변경할 수 있도록 한다.

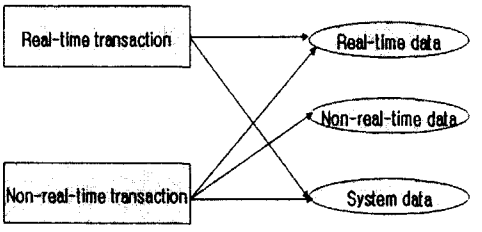


그림 5.2 보안 실시간 데이터베이스 시스템의 데이터 형태

Fig. 2.2 Data types of Secure Real-Time Database Systems

1) 실시간 데이터

실시간 데이터는 데드라인과 같은 시간 제약 조건을 갖는 실시간 트랜잭션에 의해서 요구되는 데이터이다. 이 데이터는 적시에 사용가능한 상태로 있어야만 실시간 트랜잭션의 처리가 성공적으로 완료될 수 있다. 따라서 데드라인과 더불어 또 다른 종류의 시간 제약 조건이 보안 실시간 데이터베이스 시스템 내의 데이터와 연관된다. 예를 들어, 각각의 센서 입력은 그것이 발생한 시간에 의해서 인덱스 되어지며, 일단 데이터베이스에 들어간 데이터는 특정 시간 동안에만 유효하게 된다. 이러한 개념을 수량화하기 위한 것이 데이터의 유효 기간으로서, 이 유효 기간 밖의 데이터는 실시간 트랜잭션에서 필요로 하지 않는다. 만일 유효 기간이 지난 데이터를 요구하는 트랜잭션이 있다면, 그 트랜잭션은 비실시간 트랜잭션이다.

2) 비실시간 데이터

이 형태의 데이터는 시간 제약이 없는 비실시간 트랜잭션에 의해서만 요구되는 데이터이다. 이 데이터를 구성하는 요소는 처음부터 비실시간으로 선언된 데이터와 유효 기간이 지난 실시간 데이터로 구성된다.

3) 시스템 데이터

시스템 데이터는 데이터베이스를 운영하는데 필요한 데이터로서 데이터베이스 시스템에서 자동으로 생성되며, 트랜잭션 처리 시에 항상 필요로 한다. 시스템 데이터에는 시스템 자체에 관련된 다양한 요소들 즉 데이터베이스, 테이블, 인덱스, 접근권한과 같은 정보를 포함한다.

본 연구에서 암호화 정책을 적용하는 데이터는 시스템 데이터를 제외한 실시간 데이터와 비실시간 데이터이다. 시스템 데이터는 항상 트랜잭션이 요구하는 핫스팟(hot spot) 데이터이기 때문에 이를 암호화하면 전체 시스템 효율이 저하된다. 따라서 시스템 데이터는 암호화 대상에서 제외한다. 실시간 트랜잭션이 요구하는 실시간 데이터는 암호화되어 있을 경우 복호화 처리 시간으로 트랜잭션에 부여된 데드라인을 위반할 수 있다. 따라서 보안 수준보다는 미 처리시의 시스템에 미치는 영향 평가에 의해서 암호화 기법이 적용되어져야 할 것이다. 그러나 유효기간이 지나 비실시간 데이터로 변경된 실시간 데이터의 경우는 실시간 트랜잭션이 아닌 통계나 검증을 위한 비실시간 트랜잭션에서 요구하는 경우이기 때문에 암호화 기법은 처리 시간보다 보안 수준에 의해서 선택되어져야 할 것이다.

III. 보안 가중치

3.1 보안 등급

데이터베이스에 일반적으로 적용되고 있는 다단계 보안 모델에서는 모든 데이터와 사용자에게 보안등급(security level)을 부여한다[3]. 사용자 s의 보안등급을 L(s), 데이터 o의 보안등급을 L(o)로 표시하는데, 보안등급은 두 가지의 구성요소 <레벨(sensitivity level), 범주(categories)>로 이루어진다. 이 때, 데이터와 사용자에게 부여하는 레벨을 각각 분류(classification), 허용(clearance) 레벨이라 한다.

데이터베이스 시스템이 다양한 보안등급을 가진 정보를 포함하고 가장 높은 보안등급의 데이터를 접근이 불가능한 사용자가 있을 때 다단계의 보안 필요성이 있다. 보안 등급은 *Unclassified(U)* < *Confidential(C)* < *Secret(S)* < *Top Secret(TS)*로 나누어진다. 사용자에게 할당된 보안등급은 기밀 정보를 유출하지 않을 사용자의 신뢰도를 반영한다. 다단계 데이터베이스 시스템은 다양한 보안등급을 가지는 데이터와 인가등급을 가지는 사용자를 유지하여야 한다. 가장 일반적인 경우라면 데이터베이스의 원자적 사실(fact)에 각각 보안등급을 부여하는 것이다. 이는 데이터의 접근등급과 접근을 요청하는 주체의 접근권한에 따라 데이터의 직접 혹은 간접적인 접근을 통제하는 능력을 의미한다.

보안 실시간 데이터베이스 시스템에 암호화 정책을 적용하기 위해서는 이미 데이터베이스 설계 시 분류된 실시간, 비실시간 및 시스템 데이터에 이러한 다단계 보안 모델에서 데이터에 부여하는 보안등급을 각각의 데이터에 적용해야 할 것이다. 표 3.1은 본 연구에서 보안 실시간 데이터베이스 시스템에 적용한 보안 등급이다.

표 3.1 데이터별 적용 보안 등급
Table. 3.1 Security level for Data types

보안 등급 / 데이터 분류	U	C	S	TS
Real-time data		V	V	
Non-real-time data	V	V	V	V

표 3.1에서 볼 수 있듯이 비실시간 데이터에는 4가지 보안 등급을 모두 적용할 수 있고, 실시간 데이터에는 Confidential(C)과 Secret(S) 두 등급만을 적용한다. 이러한 이유는 실시간 데이터 특성상 비분류된 데이터가 존재하지 않기 때문이다. 또한 실시간 데이터에 Top Secret(TS) 보안 등급을 적용하게 되면 암호/복호화 처리시간의 증가로 데드라인 내의 처리가 저하될 것이기 때문이다.

3.2 가중치

보안 실시간 데이터베이스 시스템 내의 트랜잭션은 여러 가지 속성에 의해서 분류된다. 첫째, 지정된 시간 제약 조건을 위반하는 경우의 영향도에 의한 분류로 하드, 펌, 그리고 소프트 시스템으로 구분할 수 있다[10]. 하드 데드라인 적용 업무에서, 데드라인을 위반한다는 것은 재앙과 같다고 볼 수 있다. 일반적으로, 하드 데드라인이 위반되면 시스템에 아주 큰 음수 값이 전달된다. 그렇지만, 펌이나 소프트 적용 업무에서 데드라인을 위반하는 것은 성능이 떨어질 뿐 큰 재앙은 입지 않는다. 둘째, 트랜잭션의 발생 형태에 의한 분류로 주기성과 비주기성 및 산발성으로 구분할 수 있다. 규칙적인 발생 시간을 갖는 트랜잭션을 주기적이라고 하고 발생 시간이 규칙적이지 않다면 비주기적이라고 한다. 일반적으로 비주기적 트랜잭션은 소프트나 펌 데드라인을 갖는다. 마지막으로 산발적 트랜잭션은 비주기적이면서 하드 데드라인을 갖는 트랜잭션을 말한다. 셋째, 데이터 접근 형태에 의한 분류로 미리 정의된 트랜잭션인지 무작위로 접근하는 트랜잭션인지의 여부를 구분한다. 넷째, 데이터 요구 조건에 의한 분류와 다섯째, 실행시간 요구 조건을 인지하는 지의 여부로 구분한다. 여섯째, 접근한 데이터 타입에 의한 분류로 연속성, 이산성 또는 둘 다 접근하는 지의 여부로 구분한다.

본 연구에서는 보안 등급과 트랜잭션 처리 결과에 따른 영향에 의해 암호화 방식을 적용하는 방안을 제시한다. 트랜잭션의 분류는 하드, 펌, 소프트 트랜잭션으로 구분하여 표 3.2에서와 같이 가중치를 부여하였다.

표 3.2에 나타나 있듯이 실시간 하드 트랜잭션으로 보안수준이 높으면 가중치 2를 보안수준이 낮으면 가중치 1을 할당한다. 이렇게 가중치를 할당하는 이유는 다음과 같다. 첫째, 하드 트랜잭션에 높은 보안 수준의 암호화 기법을 적용하면 처리시간 지연으로 인해 트랜잭션의 데드라인 내 처리가 어려울 것이기 때문이다. 둘째, 하드 트랜잭션에서 필요로 하는 주기적 또는 산발적 실시간 데이터의 특성상 데이터의 유효기간이 짧기 때문에 공격자가 해독한다고 해도 이미 그 의미가 상실된 이후가 될 것이기 때문이다.

마찬가지 이유로 펌과 소프트 트랜잭션인 경우 보안등급에 의해 각각의 가중치를 3, 4, 5, 6으로 부여하였다. 비실시간 데이터의 경우는 실시간 트랜잭

션에 의해서 요구되는 경우가 없기 때문에 암호화로 인한 처리시간 지연에 영향을 받지 않는다. 따라서 보안 등급에 의해 각각 7, 8, 9, 10이란 높은 보안 수준의 암호 방식을 적용할 수 있도록 하였다. 이 가중치에 의한 암호화 정책 결정으로 암호화로 인한 성공률 저하를 최소화하여 보안 실시간 데이터베이스 시스템의 시간성을 향상시킬 수 있다.

표 4.2 데이터와 트랜잭션 분류에 따른 가중치
Table. 3.2 Weight values for data and transactions classification

	Real-time data			Non-real-time data
	Hard	Firm	Soft	
TS	-	-	-	10
S	2	4	6	9
C	1	3	5	8
U	-	-	-	7

IV. 동적 암호화 관리자와 성능 평가

본 절에서는 시간성 향상을 위해 암호화 관리자의 구성은 그림 4.1과 같다. 암호화 관리자는 트랜잭션의 우선순위와 표 3.2에 있는 가중치에 따라 가중치를 할당하고 할당된 값에 의해서 암호 방식을 결정하는 암호화 관리자를 제안한다.

여기서 트랜잭션 관리자는 실시간 트랜잭션에 대한 스케줄링과 동시성 제어를 수행한다. 스케줄링을 위한 우선순위 할당에는 HED(Highest Earliest Deadline) 기법[11]에 보안 특성을 반영하여 적용한다. HED는 데드라인 안에 끝나는 값의 합을 최대로 하기 위한 방법으로 ED와 HV(Highest Value)를 이용하여 AED를 확장시킨 것이다. 우선순위의 할당은 트랜잭션의 가중치에 기본을 두고, 우선순위가 부여된 버켓의 계층으로 구분한다. 각 버켓은 경계값인 최대값과 최소값을 갖고 있으며, 그 경계값 내에 속하는 트랜잭션을 버켓 안에 포함하게 되고 버켓 안에서는 AED방법과 유사하게 HIT와 MISS로 구분을 하고 HIT에서는 ED를, MISS에서는 HV방법을 사용한다. 여기에 보안 수준에 따른 가중치에 따라 우선순위 할당을 변경하도록 한다. 이유는 데드라인 내의 처리에 보안에 의한 데이터 암호/복호화 시간이 추가되기 때문이다.

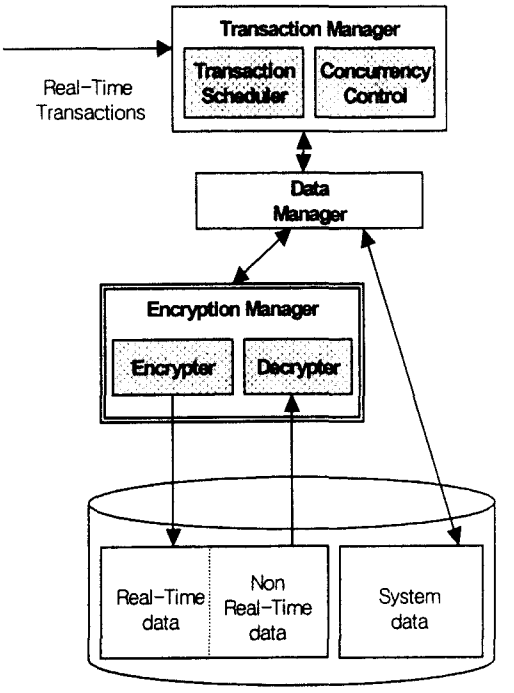


그림 4.1 암호화 관리자
Fig. 4.1 Encryption Manager

본 실험에서 적용한 스케줄러의 우선순위는 다음 공식에 의해 할당된다.

$$P_T = \begin{cases} (B_T,\ 0,\ D_T,\ I_T,\ S_L) & \text{if Group = HIT} \\ (B_T,\ 1,\ 1/V_T,\ I_T,\ S_L) & \text{if Group = MISS} \end{cases}$$

B_T : 트랜잭션 T의 버켓
V_T : 트랜잭션 T의 값
D_T : 트랜잭션 T의 데드라인
I_T : 트랜잭션 T의 ID
S_L : 트랜잭션 T의 보안레벨

트랜잭션 관리자의 동시성 제어는 보안 실시간 데이터베이스 시스템을 위해 동시성 제어 기법으로 연구된 secure 2PL, SRT-2PL 중 한 개의 데이터에 대한 두 가지 버전을 유지하여 비밀 채널 제거와 불간섭 성질을 만족시키는 SRT-2PL[12]을 적용한다.

데이터 관리자는 트랜잭션 처리 시에 필요한 데이터 중에서 시스템 데이터를 제외한 실시간 데이터와 비실시간 데이터에 대한 암호화를 암호화 관

리자에게 지시한다. 데이터 관리자는 데이터의 실시간 및 비실시간 분류 정보와 그 데이터를 접근하는 트랜잭션의 타입과 보안 등급에 대한 정보를 포함해서 암호 관리자에게 보내내다. 암호 관리자는 표 3.2의 값을 바탕으로 실시간 데이터와 비실시간 데이터에 대한 암호/복호화를 수행한다.

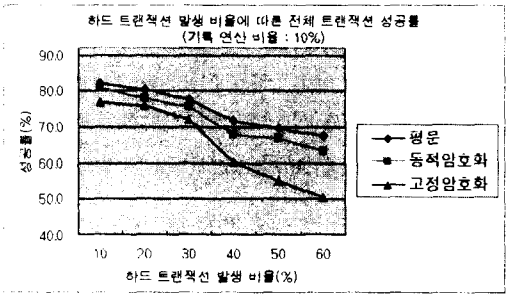


그림 7.2 트랜잭션 성공률의 비교
Fig. 4.2 Comparison of transaction hit-rate

제안한 정책을 평가하기 위해 적용된 암호화 기법은 키의 길이른 가변적으로 변경할 수 있는 RC4 알고리즘을 사용한다[7]. 보안 등급에 따라 RC4 알고리즘의 키 크기를 조정하면서 사용한다. 최고의 보안 수준인 가중치 10은 키의 크기를 128 비트로 하고 가중치가 하나씩 감소할 때마다 키의 크기를 8비트씩 줄여서 적용한다. 여기서 확인하고자 하는 것은 유동성 있게 가중치를 적용하였을 때의 실시간 트랜잭션 성공률이 고정된 암호화 기법을 적용하였을 때보다 얼마나 증가하는가이다. 그림 4.2는 128비트로 키가 고정된 암호화 방식과 유동 암호화 방식을 그림 4.1에서와 같이 보안 실시간 데이터베이스 시스템에 적용하였을 때의 성공률 차이를 보여준다. 여기서 공유 데이터의 충돌 횟수를 변화시킬 수 있는 기록 연산의 비율을 10%로 하고 트랜잭션의 성공률에 영향을 주는 하드 트랜잭션의 발생 비율을 증가 시키며 실험을 하였다.

평가 결과 보안을 유지하기 위해서는 어느 정도 트랜잭션의 성공률 저하는 감수해야 하는 것으로 나타났다. 평문을 이용한 기존의 기법에서 관찰되는 바와 같이 하드 트랜잭션이 40%이상 되었을 때 트랜잭션의 성공률이 급격히 하락하는 것을 알 수 있다. 이러한 이유는 하드 트랜잭션의 실행이 보장되어야 하기 때문에 다른 트랜잭션들이 처리 성공률이 급격하게 저하되기 때문인 것으로 연구되었다. 따라서 암호화 기법이 적용되었을 때도 마찬가지 현상이 발생하지만 고정된 암호화 기법을 적용하였을 경우는 그 영향이 더 크다는 것을 알 수 있다. 이유는 하드 트랜잭션의 실행이 보장되는 상태에서 암호화의 수행이 동적 암호화 방식보다 고정 암호화 방식이 더 많은 영향을 미치기 때문이다. 실험 결과 동적 암호화 기법을 적용하면 고정 암호화 기법을 적용한 경우보다 보안 실시간 데이터베이스 시스템의 시간성을 향상시켜 트랜잭션의 데드라인 내 처리율이 증가됨을 알 수 있었다.

V. 결 론

보안 실시간 데이터베이스 시스템에서 발생하는 데이터를 암호화하기 위해서는 구성하고 있는 데이터와 발생하는 트랜잭션의 특성을 고려하는 것이 중요하다. 따라서 본 논문에서는 데이터를 실시간, 비실시간 및 시스템 데이터로 분류하였고, 트랜잭션은 미처리 시 시스템에 미치는 영향도에 따라서 하드, 펌, 소프트로 분류하였다. 이렇게 분류한 정보와 트랜잭션의 데드라인에 의해 부과된 우선순위를 고려하여 적용할 암호화 기법을 동적으로 결정함으로써 암호화 처리로 인한 트랜잭션의 데드라인 내의 처리율 저하를 최소화하였다. 하지만 보안을 위해서는 어느 정도 성능의 저하는 감수해야 한다는 것도 알 수 있었다. 본 논문에서 제안한 방법은 평문을 사용하는 경우보다 성공률이 5% 정도 감소했지만 고정 암호 방식보다는 5%정도 높은 것을 볼 수 있다. 특히 하드 트랜잭션의 발생 비율이 40%이상인 경우에는 고정식보다 8%이상 트랜잭션 성공률이 높은 것으로 평가되었다. 따라서 제안한 암호화 관리자에 적용된 기법은 보안이 필요한 보안 실시간 데이터베이스 시스템에 유용하게 사용될 수 있을 것이다.

본 논문에서 제안한 암호화 정책은 분산 환경이 아니기 때문에 향후 분산 실시간 데이터베이스 시스템에서의 암호화 정책에 대한 연구가 필요할 것이다.

참고문헌

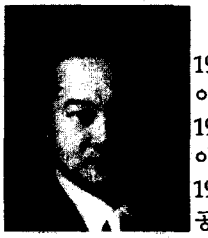
[1] Sang H. Son, Robert Zimmerman, and Jorgen Hansson, "An Adaptable Security Manager for Real-Time Transactions," 12th Euromicro Conference on Real-Time Systems, 2000.

[2] Sang H. Son, "Supporting Timeliness and Security in Real-Time Database Systems," 9th Euromicro Workshop on Real-Time Systems, 1997.

[3] S. Castano, et.al., Database Security, Addi-

son- Wesley, Reading, MA, 1994.
[4] V. Atluri, S. Jajodia, and B. George, Multilevel Secure Transaction Processing, KLUWER ACADEMIC PUBLISHERS, 2000.
[5] T. F. Lunt, Research Directions in Database Security, Springer-Verlag, 1992.
[6] I. Mavridis, G. Pangalos, and M. Khair, "EMEDAC : Role-Based Access Control Supporting Discretionary and Mandatory Features," IFIP TC11 WG11.3, 13th Working Conference on Database Security, 1999.
[7] Confirmed Test Vector and Program for RC4. on line available at http://www.qrst.de/html/ dsds/rc4.htm. 2004
[8] Rhee, Man Young, "Cryptography and secure communication," McGraw-Hill, 1994.
[9] S. J. Lee and H. Y. Bae, "Data Compression Management Mechanism for Real -Time Main Memory Database Systems," Proceedings of the 4th International Conference on Database Systems for Advanced Applications, 1995.
[10] S. H. Son and Y. K. Kim, "Predictability and Consistency in Real-Time Database Systems," Proceedings InfoScience, 1993
[11] J. Huang, J. A. Stankovic, K. Ramamritham and D. Towsley, "On Using Priority Inheritance in Real-Time Databases," Proceedings 12th Real-Time Systems Symposium, IEEE, 1991.
[12] R. Mukkamala and S. H. Son, "A Secure Concurrency Control Protocol for Real-Time Databases," IFIP TC11 WG 11.3, 9th Working Conference of Database Security, 1995.

저자 소개



이순조(Soon-Jo Lee)

1985년 인하대학교 전자계산학과 이학사
1987년 인하대학교 전자계산학과 이학석사
1995년 인하대학교 전자계산공학과 공학박사
1997년~현재 : 서원대학교 컴퓨터정보통신공학부 부교수
※관심분야 : 데이터베이스 시스템, 정보보안, GIS